SET IT UP RIGHT: John Dvorak tells you how . . . pg 14

DOS RESOURCE GUIDE

The PC Productivity Magazine

AN IDG COMMUNICATIONS PUBLICATION
NUMBER8
MARCH 1993
$4.95
CANADA $5.95

Getting CONFIG.SYS Right!

DATA REFLECTIONS:

Trick MIRROR Sees Only FAT . . . pg 27

Write PROMPTS With Personality

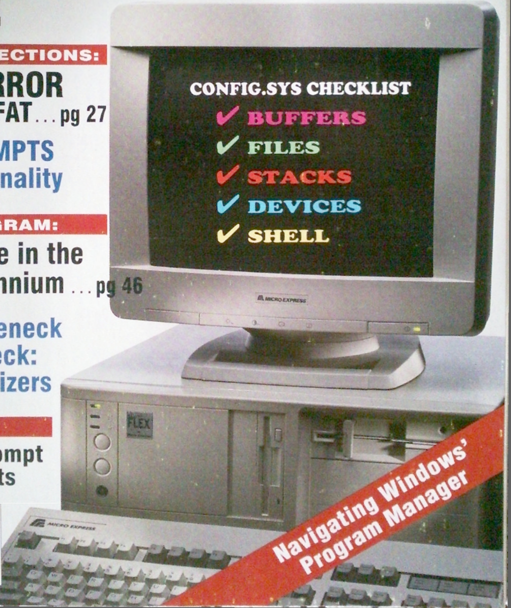
FREE PROGRAM:

Be On-Time in the Next Millennium . . . pg 46

From Bottleneck To Breakneck: Disk Optimizers

INSIDE:

8 C-Prompt Assists



Navigating Windows' Program Manager

AN IDG COMMUNICATIONS PUBLICATION
03
0 74470 78613 9

You don't have to give up DOS to get fabulous TrueType fonts!

New TRUETYPE® FOR DOS goes beyond Windows™ to give you 36 beautiful scalable fonts plus 1000's of special effects.

Who says you have to move to Windows to get great looking fonts? With new TrueType for DOS, you can create high-impact documents with WordPerfect, Word, Works, and the printer that's sitting on your desk.

You see, TrueType for DOS is a collection of 36 fonts that you can select from within your word processor and instantly scale to any size you choose—from 2 to 1000 points. Want to shrink your type down to the "fine print"? No problem. Want to blow it up so that each letter is thirteen inches tall? TrueType for DOS makes it easy.

But that's not all. With TrueType for DOS you can easily combine fonts with *thousands* of high-impact special effects including patterns, outline, shadows, and reverse effects. Over 50,000 possible combinations are always at your fingertips! This means you're suddenly free to be creative, and let's admit it—just have fun!

By the way, with TrueType for DOS, there's nothing new to learn. It installs in minutes, then works transparently with WordPerfect, Word, and Works. Your font selections are instantly scaled to size "on the fly" using the latest TrueType technology. Best of all, TrueType for DOS works with the LaserJet, Inkjet, or Dot Matrix printer you're using right now! (Other TrueType fonts you own or buy will also work perfectly with TrueType for DOS).

Why not put TrueType for DOS to the test today. We guarantee you're going to love it!

To order, see your dealer, or call

1-800-888-9078

Just **$99.95**
Comes with a 60-day moneyback guarantee!

TRUETYPE FOR DOS

by MicroLogic Software

TrueType is a registered trademark of Apple Computer. Other brand and product names are trademarks of their respective holders. Copyright © 1992 MicroLogic Software
1351 Ocean Avenue, Emeryville, CA 94608. All rights reserved.

Circle 108 on Reader Service Card

DOS RESOURCE GUIDE

EDITOR
Michael J. Comendul
SENIOR EDITOR
David Essex
EDITOR-AT-LARGE
Eric Maloney
ARTICLE EDITORS
Patricia Cummings, Beth S. Jala, Anne Fischer Lent, Marilyn McMaster
TECHNICAL EDITORS
Hardin Brothers, Mare-Anne Jarvela, John Wolfskill
CONTRIBUTING EDITOR
Dan Gookin
ART DIRECTOR/ MANUFACTURING MANAGER
Lou Ann Morin
ELECTRONIC PUBLISHING
Lindy P. Greenwood

How to contact us: *DOS Resource Guide* welcomes your comments, DOS tips, and article submissions. Submit them care of our offices at the address below, or by CompuServe: 75300,2361; or MCI Mail: 469-7260.

DOS Resource Guide (ISSN: 1056-7364) is published six times per year by Business Computer Publishing Inc., an IDG company. Business and editorial offices are located at 80 Elm St., Peterborough, NH 03458, 603-924-0100. Entire contents copyright 1992, by Business Computer Publishing Inc. No part of this publication may be printed or otherwise reproduced without written permission of the publisher. Subscriptions to *DOS Resource Guide* are available in the U.S. for $27 annually. Application to mail at Second-Class postage rates is pending at Peterborough, N.H. and additional offices. *DOS Resource Guide* makes every effort to verify its articles and program listings but assumes no responsibility for damages due to errors or omissions.

Business Computer Publishing Inc.

PRESIDENT/PUBLISHER
Jeff DeTray
EDITORIAL DIRECTOR
Michael J. Comendul
ADVERTISING SALES
Maureen Evans
National Accounts Manager
(603) 924-0269, (800) 441-4403
Linda M. Guyette
Sales Representative
(603) 924-0130, (800) 441-4403
NEWSSTAND CIRCULATION
Kemco Publishers Services
(603) 924-0224

IDG COMMUNICATIONS *DOS Resource Guide* is a publication of International Data Group, the world's leader in information services on information technology.

Are DOS Problems and PC Problems the Same?

If IBM had to redesign its original PC-compatible, would it stick video control and hardware housekeeping in the middle of the Intel chip's directly addressable space? If Microsoft had more running room, would it have limited application memory to 640K?

For better or for worse, such early decisions created DOS computing and initiated the lively debate on whether DOS is getting a bad rap for PC problems in general.

Every Tuesday morning, the editors of *DOS Resource Guide* sit around a table discussing article ideas and changes in coverage, while adding fuel to this same debate.

Think about it. When you are beefing about DOS being cantankerous, are you actually complaining about DOS, DOS software, or some unavoidable hardware problem? Take, for example, the lack of a data storage standard that makes a word processing file completely incompatible with a database program—yet they both consist of words and numbers. Haven't you tried, without success, to read a file that had been sent to you in WordPerfect format? The printed report is prettified, full of bold subheads and text that's indented from both edges, but you can't read its electronic counterpart because you're trying to call it up in an old copy of Electric Pencil. Maybe you've received a file that you'd like to merge into a document, but someone's typed it into a spreadsheet in order to take quick advantage of the spreadsheet's cells as tabbed columns.

But are these DOS problems? I don't think so.

Because incompatibilities such as the above are real and everywhere in DOS computing, Microsoft invented Windows with, among other accessories, its dynamic data exchange. But is Windows, and its autocratic way of handling hardware and software, the answer to PC ills?

Certainly, Windows is one answer, but many DOS programs make data a negotiable currency in different types of programs. Software publishers have achieved this universality in a couple of ways. One elementary method creates all-in-one programs, such as Spinnaker's Eight-In-One or Lotus Works; another method adds a utility to the mix—Mastersoft's Word for Word or White Crane's Auto Import—whose job it is to exchange data between programs. Lastly, of course, DOS programs themselves are becoming so educated that many popular word processors can import and convert foreign file formats on the fly. And it's a lean-featured spreadsheet that can't read a Lotus WK1 file.

But problems persist. After all, word processors convert files from one word processing format to another, but you'll have to till some electronic soil to get a table of numbers from your spreadsheet into your word processor. DOS can solve such problems, but the solution is tougher than it needs to be.

In future issues, *DOS Resource Guide* will try to remove some of the mysteries surrounding incompatible file types. What do you think?

Michael J. Comendul

DOS RESOURCE GUIDE

DOS

Getting CONFIG.SYS Right 14

Getting DOS off to a good start is important whether you're a beginner or an advanced user. John Dvorak gives you the Inside Track on the vagaries of adding buffers, files, and devices to your CONFIG.SYS setup file.

by John Dvorak and Nick Anis

Greetings from Your DOS Prompt 25

If DOS is giving you a blank stare first thing in the morning, add some personality to your prompt with the help of ANSI.SYS.

by Antonio and Alfonso De Pasquale

The Fairest DOS Command of All 27

The MIRROR command is DOS's first try at data protection. Its two files keep backups of your deletions and the vital File Allocation Table (FAT). One file lets you restore lost files, and the other can save you from your worst nightmare.

by Dan Gookin

BAT

Taking Care of the Environment 33

If your batch files are becoming more complicated, you may need to adjust the environment in which they operate. Adjustments to your COMSPEC, PROMPT, or PATH are handled in this issue's batch file primer.

by Hardin Brothers

FYI

How to Talk So Your Modem Will Listen 38

The Hayes AT language is not as dead as Latin or Greek, in fact, it's easier than you think to adjust the squeal of your speaker or alter the answer mode of your modem.

by John Premack

From Bottleneck to Breakneck 43

If your data access time seems stodgy to you, you may be able to change it. Using a disk optimizer regularly can unravel a tangle of disparate files and put each file back in contiguous sectors. Data placed contiguously on disk is faster and more efficient to access.

by Patrick Marshall

BAS

The Perpetual Calendar 47

For those of you who want to review the highlights of your social life, our perpetual calendar, written in QBasic, not only lets you review the last 300 or so years, but also promises to keep you on schedule well into the next millennium.

by Antonio and Alfonso De Pasquale

Understanding the Keys to QBasic 51

At the core of QBasic is a handful of commands—INKEY$, SELECT CASE, and DO...LOOP, among others—whose structures are essential to better programming.

by John Wolfskill

Number 8
March 1993

COVER PHOTO BY LARRY DUNN

Pictured on our cover is the 20Mz Micro Flex 386SX from Micro Express.

DRG

DEPARTMENTS

Letters to the Editor 4
Tips from Readers 6
Best of the Batch 8
Q&A: DOS, Disks, and Dilemmas 70
Index to Articles 73

COLUMNS

Editorial
Are you madder at Microsoft or IBM?
Michael Comendul 1

Gookin on DOS
Follow the PATH to hard disk heaven.
Dan Gookin 10

Hardware Update
Is the pen mightier than the keyboard?
Stan Miastkowski 64

New DOS Software
To see programming is to believe it.
Alan Zeichick 67

TIP

Let Your Fingers Do the Editing
For those who aren't comfortable with the mouse, the DOS 5 editor builds in a number of keyboard shortcuts.
Kay Yarborough Nelson 20

Put More Lines on Your Screen
If your monitor and color adapter can accommodate the change, DOS will let you add more lines to a screenful of text.
Jeff DeTray 22

Files to the Max
If you're using lots and lots of short files, take note of a way around directory limits.
Kay Yarborough Nelson 37

A File without a Name
Here's a trick to guarantee some filename secrecy that employs characters from the extended ASCII set.
Kay Yarborough Nelson 42

Favorable Attributes
Protect your files from inadvertent deletion: Make them read-only.
Kay Yarborough Nelson 46

XCOPY Asks for a Disk
You can use XCOPY as a backup assistant, but be warned....
Todd Martin 57

Color-Coordinated Drives
Stay organized by altering your display's color to signal a change of drives.
Kay Yarborough Nelson 62

Keeping Your Screen Clean
Tidy up after unwanted batch file messages by using a few NUL tricks.
Jeff DeTray 63

WIN

Expanding Windows: Adding a New Program 58
The master control of your Windows desktop, the Program Manager, is the key to adding, adjusting, and dumping programs and icons from this graphics interface.
by Michael Banks

DRG LETTERS TO THE EDITOR

Good Reading

The discovery of *DRG* was a very lucky one for me. Though I've taken several programming courses, I have had difficulities using the knowledge and putting it all together. Your guide has helped me very much. I have just received all the back issues. They should make good reading for several weekends to come. I most enjoy the simple programs that I can use right away.

Thank you for a useful magazine; I will continue to look for it.

George Sessions
Goleta, Calif.

DRG Fills the Hole

From the moment I browsed through *DRG*#1 at the local drug store, I knew *DRG* was important. Being a rookie at the C:> prompt, I knew there was lots of useful information in your magazine. Since then, I've checked the newsstand regularly and haven't been disappointed yet. Now I have just received my first copy in the mail. I'm enrolled in a microprocessor technician course at Algonquin College. I have made my professors aware of *DRG*, and I expect you will have them as subscribers soon. *DRG* fills a hole that many of us are extremely happy to see filled. Keep the advertising to a minimum and reader input to a maximum. Thanks for a fine magazine.

Doug Lendrum
Fallbrook, Ontario

The Best of Both Worlds

I purchased a copy of *DRG* today and wish to express my thanks to you for producing such a useful publication. Although I have "given in" and bought Windows, I still enjoy the best of both worlds as an NDOS user. Before I bought Windows, my entire library of software was accessible from a number of batch files, employing The Norton Utilities' Batch Enhancer. I still use batch files for certain maintenance tasks, such as virus scanning and disk preparation (wiping, formatting, and surface testing). I look forward to future editions of *DRG* in the University of New Brunswick's bookstore.

Neill McKay
Frederickton, New Brunswick

Talking a Different Language

I have been busy trying out new tips. Congratulations, you have a great magazine.

I ran across a small problem with the article "Once a Day" in *DRG*#1, however. I found the article interesting and proceeded to type in the program listing. It didn't work. I read the article again and was able to solve the problem, thanks to a small explanation about the importance of the name of the CURRENT.BAT file: non-English DOSes don't return a "current date is..." message to the date command!

My point is that I would like to see even more explanations about the program listings and some of the syntax that is used. For example, I have yet to see information on the ECHO command that alludes to the period or the piping sign. Most of the literature I've read only mentions the on/off switch and how to obtain a blank line in a batch file. I think it would be really interesting to have a complete description of that command, or at least, an explanation of its use in the program listing.

I think that many of your readers like to understand what they are typing, both to satisfy their intellectual curiosity and to adapt the programs to their personal needs.

Jacques Dupraz
Lancrans, France

Virus or No Virus?

I found "Protect Your Data from Computer Viruses" by Alan Zeichick [*DRG*#5, page 50] very informative. I very much enjoy broadening my awareness of DOS in particular, and computers in general. Because I routinely download files from public bulletin boards, I habitually scan my disks and hard drive for viruses. Nevertheless, I followed the advice provided by the author, ran CHKDSK, and determined that DOS could find only 654,336 bytes of total memory. I ran CHKDSK on my original DOS disks and found the same problem. According to Mr. Zeichick my "equipment was infected." When two commercial antivirus programs failed to detect a virus, I did some further investigation and learned that some non-IBM computers do not report 1,024 bytes of memory allocation for the extended BIOS data area. Con-

Let Us Hear From You . . . .

DOS Resource Guide wants to know about your computing trials and tribulations. The editors welcome your correspondence and encourage you to share your DOS tips or criticize ours.

You can reach us at ***DOS Resource Guide*, 80 Elm St., Peterborough, NH 03458,** or reach us through CompuServe: 75300,2361 sequently, it may not be totally accurate to state that anything less than 655,360 total bytes of memory (assuming 640K is installed) means trouble. Sometimes it may mean your computer came from Wal-Mart instead of IBM.

Ronald Vanover
Maineville, Ohio

OMISSIONS AND CORRECTIONS

MOVE.BAT Misunderstanding

I suspect there is a typo in Mr. Schellenger's batch file MOVE.BAT [*DRG*#6, page 8]. When I try to use this program, it copies the file to the specified directory, but ends up giving me the ERR4 message, and the source file is not deleted.

The command I've been giving is:

```
MOVE C:\DOS\MYFILE.BAT
 C:\BATCH\MYFILE.BAT
```

Is it me or the listing?

Pat Orpen
Quebec

The listing is correct as printed, but our instructions could have been better. In this rudimentary Move program, you must be in the directory in which the file to be moved resides, and rather than type complete source and target paths, the syntax is abbreviated. You need type only the filename to be moved, followed by the path and directory into which you want the file to be moved.

In the example above, once you are in your DOS directory, you have only to type:

```
MOVE MYFILE.BAT C:\BATCH
```

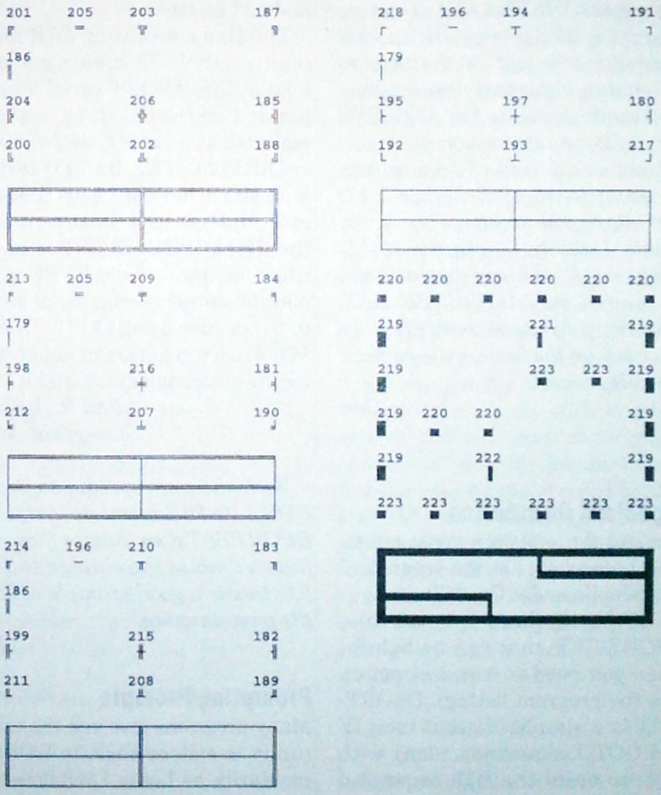
—Eds.

Re-box it!

We goofed in "Box It Up" [DRG#5, page 63]. *The programs SBOX.BAT and DBOX.BAT redefined some of the number-pad keys as single or double rules and single or double angles, using DOS's extended character set to create rudimentary drawing programs that we told you will work in DOS 5.0's editor. Unfortunately, they don't.*

The two programs do work in Edlin. DOS 5.0 Edit users can draw boxes and frames in the traditional manner, by holding down the Alt key while pressing the number-pad keys corresponding to the shape desired. See the chart below.

—Eds.



Cent Sign Problem

Help! My keyboard does not have a cent sign key, and it is a figure I often use in my business. In the ASCII table, its value is 135, and in the hex table its value is 9B. I would like to create a line in my AUTOEXEC.BAT file that will activate the cent sign each time I boot my computer. Is it possible to create a macro for this sign, then add it to my AUTOEXEC.BAT file?

Mark Goldstein
Baltimore, Md.

Your approach doesn't seem feasible. A Doskey macro will not work, because DOSKEY.COM is active only at the DOS prompt. Although it is possible to assign the character to a function key, most applications read the keyboard before ANSI.SYS can make the substitution.

A better solution is a shareware program (such as Redefine 2.17) that allows you to change one key at a time. Usually, a program of this type is small and very fast, and can be loaded in AUTOEXEC.BAT. However, you'll have to sacrifice another key on the keyboard—perhaps the tilde or caret.

—Eds.

Organized Directories

Congratulations on a great magazine. I especially like the programs I can compile under QuickBasic.

I'd like to share a batch file, DSORT.BAT, that can be helpful when you need to sort a directory (see the program listing). DSORT.BAT is a simple file that uses IF and GOTO commands, along with SET, to build the DIR command line. The GOTO in the second line jumps over the REM lines to the label START. The IF commands check to see whether you entered ? for help or failed to enter parameters. In either case, the help screen is brought up. If a third parameter (%3) is present, IF checks it for the type of output and uses GOTO to send control to the correct label.

There, IF determines the correct type of sort, then uses the SET command to build an environmental variable, named ASORT, containing the type of sort needed.

By itself, SORT starts at the name of the file and sorts alphabetically. Adding /+10 tells SORT to move 10 spaces to the right in the listing and sort by the extension. /+16 tells SORT to move 16 spaces and sort by size.

The line containing DIR and a pipe symbol (|) separates commands. The ASORT variable is expanded and sent, along with the redirection command, to the printer or DIRTEMP.TXT file. This text file is created if it doesn't exist. If it does exist, information is appended to the file. DIRTEMP.TXT is handy when you want to use TYPE to display directory information or send it to the printer using COPY. The line `SET ASORT=` releases memory used for the environmental variable.

Phil R. Atkinson
Springfield, Mont.

With the appropriate switches, DOS 5.0's DIR 5 will do everything DSORT.BAT does. But for those who haven't made the change to DOS 5.0, this is a good example of batch file programming.

—Eds.

Prompting Prompts

Many programs give you the opportunity to exit, or shell, to DOS momentarily, as Lotus 1-2-3 does. But sometimes you forget that you have a program running, and you try to start a new one, which often produces an "Out of memory" error message. But if you start the program from a batch file, you can use a prompt that lets you know that a program is running.

That batch file need contain only four commands:

```
@ECHO OFF
PROMPT Type 'EXIT' to return to
  [Program name].$_$P$G
Program name
PROMPT $P$G
```

The first PROMPT command displays the program name, performs a carriage return/line feed (_$), and displays the current directory ($P), followed by the greater than symbol ($G). The second line invokes the program you want to run. (Be sure to substitute the appropriate program name in the first two lines.) The last line restores the original prompt when you exit the program.

You can substitute any legal prompt sequence for $P$G. I use a fairly complex prompt that prints the name of the drive and directory in yellow, the commands issued in bright white, and the background in blue. This makes it easy to find the path I'm working on.

My prompt looks like this:

```
PROMPT
$E[33m$E[1m$E[44m$P$G$E[37m$E[1m$E
  [44m
```

You must first load ANSI.SYS by adding a DEVICE statement to your CONFIG.SYS file. Also, you should expand the size of the environment; otherwise, parts of the variables might be truncated.

Claude Dionne
Amos, Quebec

Fast Machine Reset

In a DOS book I found a program called RESET.COM. It worked with my old PC, but now I have a 25MHz

DSORT.BAT lets you sort directories alphabetically.

```
@ECHO OFF
GOTO START
REM DSORT.BAT sorts the directory listing by size, extension, or name,
REM sending the output to the screen, a printer, or the
DIRTEMP.TXT file
:START
  IF "%1"=="?" GOTO HELP
  IF "%1"=="" GOTO HELP
  IF "%3"=="" GOTO HELP
  IF "%3"=="print" GOTO PRINT
  IF "%3"=="screen" GOTO SCREEN
  IF "%3"=="file" GOTO FILE
:SCREEN
  IF "%2"=="name" SET ASORT=SORT
  IF "%2"=="size" SET ASORT=SORT/+16
  IF "%2"=="ext" SET ASORT=SORT/+10
  DIR %1 | %ASORT% |FIND "-" | MORE
  SET ASORT=
  GOTO END
:PRINT
  IF "%2"=="name" SET ASORT=SORT
  IF "%2"=="size" SET ASORT=SORT/+16
  IF "%2"=="ext" SET ASORT=SORT/+10
  ECHO Is printer ready...?
  PAUSE
  DIR %1 | %ASORT% | FIND "-" >PRN
  SET ASORT=
  GOTO END
:FILE
  IF "%2"=="name" SET ASORT=SORT
  IF "%2"=="size" SET ASORT=SORT/+16
  IF "%2"=="ext" SET ASORT=SORT/+10
  DIR %1 | %ASORT% |FIND "-" >>DIRTEMP.TXT
  SET ASORT=
  ECHO Your %1 listing is in DIRTEMP.TXT in current directory
  GOTO END
:HELP
  ECHO Type: DSORT [pathname of dir] [type of sort] [type of
output]
  ECHO
  ECHO Pathname of DIR same as DOS DIR command (can use wildcards)
  ECHO Type of sort    name = name, size = size of files, ext =
the extension
  ECHO Name of output  screen = view on screen, print = send
output to printer
  ECHO file = create (or append if it exists) DIRTEMP.TXT
  ECHO text file located in the current directory
  GOTO END
:END
```

End

386 PC, and it just locks up the system. Here is the Debug script from which I created RESET.COM:

```
N RESET.COM
A 100
MOV AX,40
MOV DS,AX
MOV AX,1234
MOV [0072],AX
JMP F000:E05B

RCX
10
W
Q
```

This program is helpful when you want to change the settings in your CONFIG.SYS file and reboot your system to have the changes take effect. If you can show me the correction needed for a 286, 386, and a 486 system, I would appreciate it.

William Maienschein
Laughlin, Nev.

This reset program is based solely on an instruction tucked away in an XT's ROM (and copied, out of necessity, by several clone makers).

According to the best minds in the industry, there is no software reboot method that will work on every machine in all possible circumstances. Reboots can be intercepted by 286 and 386 memory managers, can be garbled when ROM is "shadowed" to RAM, and can be a real pain when DesqView, Windows, or OS/2 are running. But you might try changing the seventh line in the script program to read:

```
JMP FFFF:0000
```

This modified version of RESET.COM created from this script should work on many machines—most of the time.

—Eds.

The Colorful Batch File

Most DOS users who write batch files are more concerned with function than with form. As long the batch file works, who cares what it looks like?

But if you're writing batch files for others, you should be concerned with presentation. Users welcome any efforts to make screens more professional and easier to read. This is true particularly when you're presenting screens of information or menus.

The batch file, EXAMPLE.BAT, is a simple template you can integrate into any batch file that presents text (see the program listing). As written, my batch file gives you a screen with a blue background and a centered green box for text. The text comprises a list of menu options, each in black type with the first letter highlighted in white. I also have added a shadow along the right and bottom edges of the box to give it a 3-D effect. (When you type in EXAMPLE.BAT, do not include the line numbers. They are for reference only.)

The code illustrates several techniques that you might find useful for other batch files. First, it uses escape sequences to give the screen color. Second, it uses other escape sequences to move the cursor, add the highlighted first letters to the menu entries, and add the shadow.

TEXT ATTRIBUTES

Number	Attribute
0	All attributes off
1	Bold on
4	Underscore (monochrome only)
5	Blink on
7	Reverse video on
8	Concealed on

Batch File ESCapades

To use escape sequences, you must install DOS's ANSI.SYS device driver. You do this by including the line `DEVICE=ANSI.SYS` in your CONFIG.SYS file. (Be sure to include the proper pathname for ANSI.SYS if it isn't in your root directory.) An ANSI escape sequence is a string made up of the escape character, the left-bracket ([) character, and a series of letters and numbers. These alphanumeric characters make up a code that ANSI.SYS uses to manage a keyboard or display function.

The escape sequence for adding colors is `ESC[`*`bg#`*`;`*`fg#`*`m`, where *bg#* is a number that sets the background color, *fg#* is a number that sets the foreground color, and *m* is a code identifying the string as one that sets the screen colors. For example, `ESC[44;37m` sets the background to blue and the foreground to white.

COLOR CHOICES

FOREGROUND COLORS		BACKGROUND COLORS	
Number	Color	Number	Color
30	Black	40	Black
31	Red	41	Red
32	Green	42	Green
33	Yellow	43	Yellow
34	Blue	44	Blue
35	Magenta	45	Magenta
36	Cyan	46	Cyan
37	White	47	White

The escape sequence for moving the cursor is `ESC[`*`row`*`;`*`column`*`H`, where *row* and *column* are the row and column numbers and *H* identifies the string as one for repositioning the cursor. For example, `ESC[06;24H` moves the cursor to the sixth row and the 24th column.

How you create the escape character will depend on what text editor you use to write the batch file. If you are using DOS 5.0's Edit, press Ctrl-P and then the Esc key. Follow this arrow with a left bracket ([).

If you are using Edlin, press Ctrl-V and follow it with a left bracket ([).

If you are using a word processor, refer to its documentation for instructions.

Looking at EXAMPLE.BAT

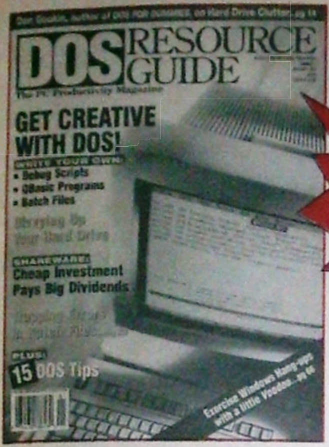
Here's a line-by-line explanation of EXAMPLE.BAT:

- Lines 1–6—Turn echoing off and set the background color to blue and the foreground color to white.

  If you want colors other than blue and white, replace the numbers in the escape sequence with the ones that represent the colors you choose. The table "Color Choices" lists your options.
- Line 7—`ESC[06;24H` moves the cursor to row 6, column 24. A white bar—the top of the text box—is drawn using the ASCII character 220. You get this character in Edit and Edlin by pressing Ctrl-Alt-220.
- Line 8—Changes the screen colors to white characters on a green background.
- Lines 9–22—Each line moves the cursor down one row, staying in column 24, and then displays text or, in lines 11 and 18, graphics. The straight lines are created in Edit and Edlin by pressing Ctrl-Alt-196.

To modify this portion of the batch file, simply replace the text presented here with your own. If you want graphics other than a line, replace the ASCII code in the Ctrl-Alt sequence with the code that represents the character you want. For instance, a double line would be Ctrl-Alt-205; a wavy line would be Ctrl-Alt-247. (Your DOS manual should include a table of ASCII characters.)

SUBSCRIBE TODAY!


Dan Gookin, author of DOS FOR DUMMIES, on Hard Drive Clutter...pg 14
DOS RESOURCE GUIDE
The PC Productivity Magazine
GET CREATIVE WITH DOS!
WRITE YOUR OWN:
• Debug Scripts
• QBasic Programs
• Batch Files
SHAREWARE:
Cheap Investment Pays Big Dividends
PLUS:
15 DOS Tips
Exercise Windows Hang-ups with a little Voodoo...pg 66

BUCK THE TREND! STAY WITH DOS!

☐ **YES!** Send me 6 Issues of DOS Resource Guide for $23.70.*
That's a DOLLAR OFF EACH ISSUE on the newsstand!

☐ **YES!** SAVE EVEN MORE ON 12 ISSUES! Send me 12 issues for $44.00.*

SUBSCRIBE TODAY!
SAVE UP TO 25% OFF THE COVER PRICE

NAME ____________________
ADDRESS ____________________
CITY ____________________ STATE ______ ZIP ______

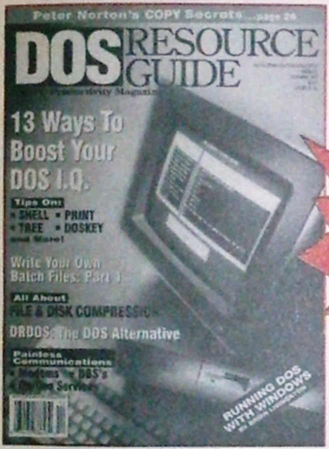
(Check, Money Order, or Credit Card—Payment must accompany your order)

☐ AMEX ☐ VISA ☐ MASTERCARD ☐ DISCOVER

CREDIT CARD # ____________________
EXPIRATION DATE: ____________________
SIGNATURE: ____________________

ID8C1

*All foreign subscriptions must be paid in U.S. funds drawn on a U.S. bank.
*Canada and Mexico: $29.70 (includes GST)/2 years $55.00
*Maryland residents add 5% sales tax.
*All other foreign (Surface Mail): $41.70 /2 years $78.00
*Foreign Air Mail: $53.70/2 years $99.00



BUCK THE TREND! STAY WITH DOS!

☐ **YES!** Send me 6 Issues of DOS Resource Guide for $23.70.*
That's a DOLLAR OFF EACH ISSUE on the newsstand!

☐ **YES!** SAVE EVEN MORE ON 12 ISSUES! Send me 12 issues for $44.00.*

SUBSCRIBE TODAY!
SAVE UP TO 25% OFF THE COVER PRICE

NAME ____________________
ADDRESS ____________________
CITY ____________________ STATE ______ ZIP ______

(Check, Money Order, or Credit Card—Payment must accompany your order)

☐ AMEX ☐ VISA ☐ MASTERCARD ☐ DISCOVER

CREDIT CARD # ____________________
EXPIRATION DATE: ____________________
SIGNATURE: ____________________

ID8C2

*All foreign subscriptions must be paid in U.S. funds drawn on a U.S. bank.
*Canada and Mexico: $29.70 (includes GST)/2 years $55.00
*Maryland residents add 5% sales tax.
*All other foreign (Surface Mail): $41.70/2 years $78.00
*Foreign Air Mail: $53.70/2 years $99.00

HELP US OUT: READER SURVEY INSIDE...PAGE 81
DOS RESOURCE GUIDE
The PC Productivity Magazine
THE ONE-STOP DOS PRIMER
The 10 Worst DOS Commands
DEBUG Almost Defined
15 DOS TIPS
PROTECT YOUR DATA FROM VIRUS ATTACK

BUCK THE TREND! STAY WITH DOS!

☐ **YES!** Send me 6 Issues of DOS Resource Guide for $23.70.*
That's a DOLLAR OFF EACH ISSUE on the newsstand!

☐ **YES!** SAVE EVEN MORE ON 12 ISSUES! Send me 12 issues for $44.00.*

SUBSCRIBE TODAY!
SAVE UP TO 25% OFF THE COVER PRICE

NAME ____________________
ADDRESS ____________________
CITY ____________________ STATE ______ ZIP ______

(Check, Money Order, or Credit Card—Payment must accompany your order)

☐ AMEX ☐ VISA ☐ MASTERCARD ☐ DISCOVER

CREDIT CARD # ____________________
EXPIRATION DATE: ____________________
SIGNATURE: ____________________

ID8C3

*All foreign subscriptions must be paid in U.S. funds drawn on a U.S. bank.
*Canada and Mexico: $29.70 (includes GST)/2 years $55.00
*Maryland residents add 5% sales tax.
*All other foreign (Surface Mail): $41.70/2 years $78.00
*Foreign Air Mail: $53.70/2 years $99.00

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 73 PETERBOROUGH, NH

POSTAGE WILL BE PAID BY ADDRESSEE

DOS RESOURCE GUIDE
PO Box 802
Peterborough NH 03458-9971

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 73 PETERBOROUGH, NH

POSTAGE WILL BE PAID BY ADDRESSEE

DOS RESOURCE GUIDE
PO Box 802
Peterborough NH 03458-9971

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 73 PETERBOROUGH, NH

POSTAGE WILL BE PAID BY ADDRESSEE

DOS RESOURCE GUIDE
PO Box 802
Peterborough NH 03458-9971

YES! Save me MONEY

Send me 12 Issues (two years) of DOS Resource Guide for $44.00.

OR

6 Issues (one year) of DOS Resource Guide for $23.70.

- Line 23—Sets the colors to black on blue.
- Line 24—Draws a black shadow line using the ASCII character 223 (in Edit and Edlin, Ctrl-Alt-223).
- Lines 25–27—Draw the vertical shadow along the right edge of the text box, starting at the seventh row of column 59 and proceeding to the 20th row of column 29. The first graphics character is ASCII 220; the others are ASCII 219.
- Line 28—Sets the colors to white on green. The number 1 before the *m* sets the text attribute to bold.

You can use attributes other than bold (see the table "Text Attributes"). Replace 1 with the number that corresponds to your choice.

- Lines 29–30—Each escape sequence moves the cursor to the first letter of each line of text in the text box and retypes the letter in bold white.
- Line 31—Moves the cursor to the bottom-left corner of the screen.
- Line 32—Leaves the batch file and changes your prompt so that it displays "Select a letter and press Enter:". The sequence `ESC[44;37;1m` ensures that the prompt color matches the background color. The sequence `ESC[K` erases to the end of the line; this ensures that the entire line is blue.

Keep in mind that while the prompt "Select a letter..." looks as if it's part of the batch file screen, it's not. Line 32 ends the batch file and takes you back to the DOS prompt, in the process changing the prompt message. If you were to type a DIR command instead of a menu choice, you'd get a directory.

For this sample menu to work, each choice in the batch file's text box must relate to a corresponding batch file that runs that program. A batch file named X.BAT would change to the XyWrite subdirectory and load XyWrite. To return to the main menu, X.BAT would need to finish with a line that ran the menu batch file. These batch files can be located anywhere, as long as their directory is in your computer's PATH statement.

Also, keep in mind that your DOS prompt has been changed. If you want to restore the DOS prompt set by your AUTOEXEC.BAT file when you turn on your computer, you'll need to reenter the prompt or put it into its own batch file. For example, if your DOS prompt is set with the command `PROMPT=$P$G`, put this command in a batch file called, say, P.BAT. Typing *P* will then reset your DOS prompt to its original state.

James R. Burridge
Thomaston, Maine

EXAMPLE.BAT. To make EXAMPLE.BAT a working menuing program, you must add your own application and utility names to the options on the menu and then extend the batch file to include complementary batch files that start each EXE file mentioned. (The line numbers in EXAMPLE.BAT are for clarity only; ignore them as you type.)

```
 1  @ECHO OFF
 2  REM   **  EXAMPLE.BAT
 3  REM  **  Main Menu - to call all MENU.BAT screens  **
 4  ECHO ESC[44;37m
 5  CLS
 6  ECHO ESC[44;37m
 7  ECHO ESC[Ø6;24H‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹
 8  ECHO ESC[42;3Øm
 9  ECHO ESC[Ø7;24H
1Ø  ECHO ESC[Ø8;24H                    MAIN  MENU
11  ECHO ESC[Ø9;24Hƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
12  ECHO ESC[1Ø;24H              XyWrite
13  ECHO ESC[11;24H              Quattro
14  ECHO ESC[12;24H              Basic (QBasic)
15  ECHO ESC[13;24H              PC Tools
16  ECHO ESC[14;24H              Reflex
17  ECHO ESC[15;24H
18  ECHO ESC[16;24Hƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
19  ECHO ESC[17;24H              Command Line... DOS
2Ø  ECHO ESC[18;24H              Reboot
21  ECHO ESC[19;24H
22  ECHO ESC[2Ø;24H
23  ECHO ESC[44;3Øm
24  ECHO ESC[21;24H ﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁﬁ
25  ECHO ESC[Ø7;59H‹ESC[Ø8;59H¤ESC[Ø9;59H¤ESC
  [1Ø;59H¤ESC[11;59H¤ESC[12;59H¤
26  ECHO ESC[13;59H¤ESC[14;59H¤ESC[15;59H¤ESC[16;59H¤ESC
  [17;59H¤ESC[18;59H¤
27  ECHO ESC[19;59H¤ESC[2Ø;59H¤
28  ECHO ESC[42;37;1m
29  ECHO ESC[1Ø;33HXESC[11;33HQESC[12;33HBESC[13;33HP
3Ø  ECHO ESC[14;33HRESC[17;33HCESC[18;33HR
31  ECHO ESC[24;1H
32  PROMPT ESC[44;37;1mSelect a letter and press
  Enter:ESC[K
```

End

Creating a PATH of Gold

I have found PATH nirvana. Since 1983, when the PATH command debuted in DOS 2.0, I've been asking myself which directories I should put in the path, a neat mechanism that lets DOS find programs called from outside their native directories.

With the help of batch files, I pared down an originally long path to three directories. But recently I discovered a trick that lets me keep only one directory on the path and makes my system run blindingly fast. I'd like to pass on the tip, but first a detailed history lesson is in order.

> ***Clever use of DOS's PATH command makes it easier to run important programs without wasting time or resources.***

A Walk Down the Path

Managing files and programs on a hard drive is easier if you stuff everything into its proper subdirectory. You create a directory for DOS, one for WordPerfect, another for The Norton Utilities, Fastback, your communications program, and so on. Eventually your hard drive is bustling with dozens of subdirectories and sub-subdirectories. This makes everything organized and keeps your head from exploding when you need to find a key file.

The problem with having directories all over creation is that you either need to constantly type CD-this and CD-that before running a program, or type in a full pathname. For example, every time you formatted a disk, you'd have to type the full pathname to the FORMAT command:

```
C:\DOS\FORMAT A:
```

And if you wanted to start WordPerfect and load the POEM07 file, you might end up typing something like the following:

```
C:\WP51\WP C:\PROJECTS\POETRY\POEM07
```

Sure, it works, but aren't computers supposed to make life easier? Typing in such long commands definitely is not my idea of a simplified existence.

The solution is to create a search path using DOS's PATH command (see the sidebar “How to Use the PATH Comand”). The search path is a list of directories, specifically those directories that contain program files. Whenever you type in the name of a program, DOS looks in the current directory for the program file. If the program isn't found, DOS continues scanning the directories listed in the search path.

If you have a directory that contains a program you run a lot, such as WordPerfect, it makes sense to put that directory in the search path. That way, you can type *WP* to run WordPerfect without having to CD-this or CD-that or type a long, complex pathname. DOS will always find WordPerfect, because its directory is in the search path. (In

BY DAN GOOKIN

fact, some Windows applications *must* be in the path).

You typically add a PATH command to your AUTOEXEC.BAT file, though you can set or reset the path at any time at the command prompt.

PATH is followed by a space, then the names of one or more subdirectories. If more than one subdirectory is specified, you must separate their names with semicolons. For example, the following command:

```
PATH C:\DOS;C:\WP51
```

places the C:\DOS and C:\WP51 directories in the search path. This means DOS can find and run programs in those directories from any place on your system. No hard-drive-equipped PC should be without a search path.

The Path Problem

You can go overboard with the PATH command, though. Obviously, if you're well organized and have programs in separate subdirectories all over the hard drive—and you often use those programs—it makes sense to put all those directories in the path. But too many directories in the search path can bog down your system. For example, suppose you have the following eight directories in the path:

```
C:\DOS
C:\UTIL
C:\NORTON
C:\GAMES
C:\WP51
C:\123
C:\SCOM
C:\PCTOOLS
```

Right away, you'll notice that this list requires a long, complex path. I've listed each subdirectory on a line. On your screen, everything would be glued together with semicolons to form one long line of scrambled eggs—not a pretty thing to see. (Refer to the sidebar "Clearing the Path" for an interesting batch file that neatly lists the directories in your path.)

Such a long search path (and I've seen longer ones) causes DOS to overwork your hard drive looking for programs. Suppose, for example, you type in *PCSHELL* to run PC Tools' shell program. First DOS looks in the current directory, but to no avail. Then it turns to the search path and looks in C:\DOS for PCSHELL. No luck. Then it looks in C:\UTIL, C:\NORTON, and on down the search path. Finally it comes to C:\PCTOOLS and loads the program. Even on a fast hard drive, that kind of extra overhead can make you wait a few more seconds for the program to run. And if you make a typo, you'll have to wait for the "Bad command or file name" error message—and longer still to eventually run your program. The problem grows worse the longer your search path becomes.

To avoid ending up with a long path, it helps to keep only your "gang" directories in the search path. For example, DOS contains dozens of program files. A general utility subdirectory, called \UTIL, may contain a few dozen handy utilities. And \GAMES may contain all your game programs. It makes more sense to keep these directories in the path than to mention directories, such as C:\WP51 or C:\123, that may contain only one program file. Yet there's a gang directory solution for those types of directories, as well.

The Minimum Path

Your search path should contain three or four directories, tops. Anything more and DOS wastes time searching your hard drive. The solution for directories containing solitary programs is writing batch files to run those programs, then placing all those batch files in a common \BATCH subdirectory. This is what I call "the \BATCH solution."

For example, you can write a simple, nobrainer batch file to run WordPerfect, another for 1-2-3, and more for other programs on your hard drive. Toss everything into a common \BATCH directory and you can pare down an eight-directory search path to a nifty path containing just two or three directories:

```
C:\DOS
C:\UTIL
C:\BATCH
```

HOW TO USE THE PATH COMMAND

Syntax:
PATH
PATH *pathname* [;*pathname...*]
PATH ;

By itself, the PATH command displays the current search path.

When followed by one or more pathnames, PATH sets the search path to include those directories. The pathname should include the drive letter and all "parent" directories. Do not end a pathname with a backslash. If you use more than one pathname, separate it from the previous pathnames with a semicolon.

When a sole semicolon follows PATH, the search path is reset to nothing. This is the way DOS starts unless you use PATH in AUTOEXEC.BAT or at the command prompt to set a new path.

This cuts down on the time it takes DOS to find files. Instead of hunting through eight or more directories, DOS can quickly find what it needs in three directories. Also, all three directories contain an abundance of program files; DOS won't waste time looking through a series of single-program directories.

Another advantage to the \BATCH solution is that you can change or modify the search path in a batch file. For example, if you run a program that absolutely must have its directory in the path, you can change the path in the batch file you use to run the program—then change it back. The object is to keep the path short, while accommodating programs that insist on having a longer path.

Using batch files in a common \BATCH directory has a minor drawback. Because DOS reads batch files from your hard drive a line at a time, they might slow down your system. For example, running a complex batch file may take more time than having DOS search through several directories in the path. I have a solution to this problem, too—one that actually lends itself to the briefest path possible: Use part of your system's memory to act as a drive, a so-called RAM drive.

CLEARING THE PATH

Using the PATH statement too liberally can produce a hard-to-read string of subdirectory names, backslashes, and semicolons in your AUTOEXEC.BAT file.

Here's a remedy. The following batch file, SHOWPATH.BAT, displays all the directories on DOS's search path in a nice, orderly manner:

```
@ECHO OFF
ECHO Here are the directories on DOS's search path:
FOR %%X IN (%PATH%) DO ECHO %%X
```

You can type it in using the COPY CON command or DOS's text editors, EDLIN and EDIT. Then save the file to disk, preferably in your BATCH directory.

SHOWPATH.BAT uses the %PATH% variable and the FOR command to list all the subdirectories on your search path. The %PATH% variable translates into the directories on your search path when the batch file runs. Because each directory is separated by a semicolon, and the FOR command uses the semicolon as a separator, the batch file reads and displays each directory on the path on a separate line using the ECHO command.

—D.G.

The Bare Minimum

The best way to improve the speed at which a batch file executes is to copy it to a RAM drive. DOS still must access the batch file a line at a time, but because the batch file is sitting in RAM, there's no mechanical overhead. Batch files stored in RAM run almost instantaneously.

Batch files don't take up much space, so setting up a 32K RAM drive in expanded or extended memory will suffice. Adding the following command to CONFIG.SYS does the trick:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 32 512 128 /E
```

This line creates a RAM drive in extended memory, but you can use /A instead of /E to place it in expanded memory (see the sidebar "Creating a RAM Drive"). The 512 sets the RAM drive's sector size at 512 bytes, but more importantly, the third number sets the number of directory entries at 128. Without it, your RAM drive would be full after only 64 files were copied. Setting the number to 128 gives you room on the hard drive for as many as 128 batch files.

After the RAM drive is created, you can copy all your batch files to it when DOS first starts. Just stick the following line in AUTOEXEC.BAT:

```
XCOPY C:\BATCH F:
```

The above command assumes that drive F is the RAM drive. (Remember to save any new batch files you create to the hard drive, then copy them to the RAM drive.) Then you put *F:* in the path instead of *C:\BATCH:*

```
PATH C:\DOS;C:\UTIL;F:\
```

This is a fast solution, but there is still some untapped potential: Instead of copying just the batch files to the RAM drive, why not copy everything? You can, for example, write a batch file that copies all the necessary DOS files, utilities, and batch files to a large RAM drive. Provided you have enough extended or expanded memory, this makes for an excellent, ultimate solution to the search path crisis—my PATH nirvana.

On my system, I have a special start-up batch file that copies to drive F a collection of 30 or so common DOS programs, and all the files in my \UTIL and \BATCH directories. Drive F, a 2MB RAM drive, holds some 161 files; the command to create it is as follows:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 2048 512 256 /E
```

Because I'm swimming in extended memory, I can get away with a 2MB RAM drive. (If you're using Stacker, you can create a 1MB RAM drive; Stacker effectively doubles its size to 2MB. Refer to the Stacker manual.) Note the value 256, which reserves room for as many as 256 files. But don't dwell on the details. When you have a huge RAM drive, and you copy your DOS, utility, and batch file programs to it, you can list just one directory in your path:

```
F:\
```

That's it!—the full search path on my system. All the vital files are on RAM drive F. And, as a bonus, everything is accessed quickly, because it all sits in memory.

Obviously, you can't put everything on the RAM drive. MS-DOS 5.0's files occupy some 2.5MB of hard drive space, so only a handful of its commands are on the RAM drive. Everything else must be accessed via batch files, which means I had to write a few of them to make this system work.

It takes a bit of extra effort, I admit, but if you have the memory to spare, this approach lets you set up your PC to run with only one directory on the path—a single RAM drive. This has to be truly the best solution to the search path problem. (If you have a better one, write in. I'd love to hear about it.) ■

Dan Gookin is perhaps best known as the author of DOS for Dummies *(IDG Books).*
He lives in Issaquah, Wash.

CREATING A RAM DRIVE

You must use the following command line format to create a RAM drive in memory:

```
pathname\RAMDRIVE.SYS [size sector entries] [/E|/A]
```

The size indicates the size of the RAM drive in kilobytes (K). If size isn't specified, a 64K RAM drive is created.

The sector value indicates the size of the RAM drive's sectors, either 128, 256, or 512 bytes. I recommend a sector setting of 512. If you have DOS 3.x and earlier, you must specify this value (rather than accepting the default of 128 bytes). You also may state an entries value, which determines the number of files you can have in the RAM drive's root directory. The entries default is 64. For creating a RAM drive to hold all your start-up batch files or all those files plus necessary DOS files and utilities, I recommend a higher value: 128, 256, or more.

Use the optional /E switch to place the RAM drive in extended memory. The /A switch creates the RAM drive in expanded memory. Without /E or /A, DOS puts the RAM drive in conventional memory.

DOS assigns each new RAM drive the next highest drive letter in your system. For example, if you have hard drives C and D, RAMDRIVE.SYS names your RAM drive as drive E. You can place RAMDRIVE.SYS in CONFIG.SYS more than once to create multiple RAM drives. (Just make sure you don't run out of memory!)

Finally, you should know that some versions of DOS (PC-DOS and Compaq DOS, among them) call the RAM drive device driver VDISK.SYS. Its options and settings are identical to those used in RAMDRIVE.SYS.

—D.G.

Getting CONFIG.SYS Right

Whether you're a beginner, intermediate, or advanced DOS computer user, you'll appreciate having the inside track on the workings of CONFIG.SYS.

by John Dvorak and Nick Anis

You've probably noticed that once they are turned on, most PCs end up at a C:> prompt (or an A:> prompt, if there's no hard disk). You also may have noticed that some systems *don't* end up there. Maybe they display a menu or start a multitasking program. Some jump into Microsoft Windows or the DOS shell.

If you watch the screen, you'll also see that every system seems to have a slightly different route for getting where it's going when it starts up. It makes sense if you think about it. Between the time you wake up in the morning and the time you start work, you do a lot of things to get ready. Other people are doing many of the same things—but a little differently. Starting your PC works the same way. Although the general process may be the same for every computer, the steps along the way can be quite different.

DVORAK Osborne McGraw-Hill
DVORAK'S
INSIDE TRACK
TO
DOS & PC
PERFORMANCE
JOHN C. DVORAK & NICK ANIS
• Covers All DOS Versions
Includes One 3.5-Inch Disk Packed with John Dvorak's Favorite Utilities
INCLUDES DISK

The variations in a computer's route to the C:> prompt or DOS shell are made possible by your CONFIG.SYS and AUTOEXEC.BAT files. In this article, we look at the CONFIG.SYS file and the commands at your disposal to create it. To assist you in finding the best start-up routine for your system, we include some sample CONFIG.SYS files for computers with varying setup needs.

Form and Function

CONFIG.SYS is a text file stored in the root directory of the disk from which your system boots. You can edit it with any text editor or any word processing program that can create ASCII files. If CON-

Editor's Note: *The following article is adapted from Chapter 5, "Starting Your Computer," of* Dvorak's Inside Track to DOS and PC Performance *by John Dvorak and Nick Anis, an Osborne / McGraw-Hill book, copyright 1992. Reprinted with permission of the publisher. All rights reserved.*

FIG.SYS doesn't exist, DOS starts anyway; it simply uses default values for the parameters you can specify in CONFIG.SYS.

CONFIG.SYS has three functions:

- It lets you set certain parameters for DOS, such as the maximum number of files that programs in your system can use at any one time and the number of file buffers.
- It lets you load driver programs to control physical devices connected to your computer (such as a mouse or a networking board) or perform other functions not provided by DOS itself.
- It lets you specify the command processor once DOS initialization is complete.

Because CONFIG.SYS is processed only when the system starts, changes you make to the file take effect only after you reboot.

The basic syntax for entries in CONFIG.SYS is as follows:

```
name=value
```

An entry's name must begin in the first column of the line, as in this command:

```
FILES=20
```

If, during your setup, you specify an invalid directive, DOS reports an "Unrecognized command in CONFIG.SYS" error message and continues processing the rest of the file. Because DOS does not stop when it encounters an error in CONFIG.SYS, you can deactivate a directive temporarily by prefacing it with a character such as a period or an asterisk, which, in turn, makes the directive invalid.

Beginning with DOS 4.0, you can insert comments in a CONFIG.SYS file by beginning the line with REM, as in the following example:

```
REM  Set the number of files to 20
```

The entries in CONFIG.SYS are not necessarily processed in the order they appear. DOS has its own sequence for processing CONFIG.SYS directives while initializing your system, and it executes them in that order regardless of the order in which you place the directives in the file. For example, DOS always processes SHELL last, no matter where it appears in the file. The partial exception to this rule is the DEVICE directive, which can appear several times in a CONFIG.SYS file. When it comes time for DOS to look for a DEVICE directive, and it encounters more than one such line, it processes the DEVICE lines in the order in which they occur in CONFIG.SYS.

DOS Parameters

CONFIG.SYS lets you set certain parameters for DOS. The parameters you can specify are listed alphabetically in the remainder of this section; all are available in DOS 2.0 and later, unless otherwise noted.

BREAK=On
BREAK=Off

BREAK sets extended Ctrl-Break checking on or off. If BREAK is on, DOS checks for a Ctrl-C or Ctrl-Break key press every time it is called by a program for any purpose (which is quite often in most programs). If BREAK is off (which is the default), DOS makes this check only while performing input or output to a character device, such as the screen, the keyboard, a communications port, or a printer. Turning BREAK on slows DOS down slightly but makes it easier to stop the operation of programs that don't do much character I/O or don't use DOS to do it. On most systems, the increased control you get from setting BREAK on is well worth the infinitesimal performance degradation it causes.

You also can turn BREAK on and off in a batch file or at the command prompt, using the DOS BREAK command:

```
BREAK=ON
```

Turning BREAK on in CONFIG.SYS offers no particular advantage or disadvantage over invoking it in AUTOEXEC.BAT or elsewhere.

BUFFERS=*nn*

If you are using DOS 5.0, the number of buffers you have really isn't a concern. DOS 5.0 sets aside the buffer space it needs, so you don't need to set up a specific number of buffers. If you don't have DOS 5.0, read on! (For a new slant on BUFFERS, check out the sidebar "READ.ME: A BUFFERS Substitute?".)

BUFFERS specifies the number of one-sector (512-byte) buffers that DOS versions prior to Version 5.0 should set up. A buffer is simply an area of memory that DOS uses when a program transfers information to and from the disk. The allowed range is from one to 99 buffers; the default is two buffers for PC and XT systems and three for AT, 386, and 486 systems. If you have DOS 4.0 and later, you can use the /X parameter to set up a buffer in expanded memory or create secondary, or read-ahead, buffers, which are useful with slow hard drives. If you have more than 512K of RAM and use DOS 3.3 or later, the buffers setting defaults to 15 for any system, unless you override this setting by adding a `BUFFERS=nn` line, such as the following, to your CONFIG.SYS file:

```
BUFFERS=20
```

Device drivers for certain hard drives, such as SCSI drives with large DOS partitions, may increase the buffer size. Increased buffer size reduces the amount of memory available for other purposes, such as running programs. You'll find a disk cache a better alternative than adding buffers for these situations.

To understand why buffers are necessary, consider the following scenario. Your spreadsheet program needs to read 10 bytes from your worksheet file, beginning at byte number 230. But the disk is divided into 512-byte sectors, and the disk hardware can transfer only 512 bytes at a time. This means DOS has to read the entire first sector of the file to retrieve those 10 bytes and make them available to the spreadsheet program. It uses a buffer to hold the data it retrieves.

When accessing the disk, DOS checks all its buffers first to see if the information needed already is in memory. If not, the data in the least recently used buffer is written to disk, if necessary, and that buffer is used to store the new information. (This common method of buffer allocation is typically called the LRU method, for *least recently used*.)

To control the number of sectors on the disk for which DOS looks ahead, you specify secondary buffers using a command such as this one:

```
BUFFERS=20,4
```

Increasing the number of buffers makes it more likely that the information you need from the disk already will be in a buffer. This can speed program operation significantly. It takes only microseconds to access data from memory; finding data on a disk can take anywhere from several milliseconds on a hard disk to a couple of seconds on a floppy disk. On the other hand, each buffer uses 512 bytes of memory. So, using a large number of buffers can reduce dramatically the space available for your applications. In addition, if the number gets high enough, the overhead of managing the buffers actually can require more time than it would take to get data from the disk. Buffers may be loaded into high memory using the /X switch.

If your PC has only a floppy drive and less than 512K of memory, the default setting of two buffers (three on AT and 386 systems) is sufficient. If you have a hard disk, you need more; how many more depends on the programs you're running. In most cases, 10 or 20 is sufficient. If you increase the number of files (discussed later in this section) above 20, increase the number of buffers so that you have nearly as many buffers as files.

The best way to find a good setting is to try running your system for a while with five buffers or so. (Test your system using programs that access the disk a lot and that also use a large number of files simultaneously.) Then, increase the setting to 10 and run those programs again. Next try running them with a setting of 15, and so on. Assuming you can spare the memory, the number to use is the number above which you notice no significant performance improvement.

Programs that access data sequentially (such as a word processor, which usually reads in long sections of a document at once) benefit less from additional buffers than databases and other programs that access data at random. The process of sequentially reading the data often fills up all the buffers, perhaps several times over, before there is a second request for a sector that was read earlier. By the time the second request occurs, the required sector isn't in a buffer any more. On the other hand, programs such as database managers, which go back and forth frequently between index and data files, are very likely to require repeated access to the same sector; so, additional buffers can be very helpful. But your best bet is to get DOS 5.0 so that you don't have to worry about buffer size.

If you're using a disk-caching program that supports floppy drives and hard drives, such as Multisoft's Super PC-KWIK, DOS's SMARTDrive, Central Point Software's PC-Cache, or Golden Bow's Vcache, you need to think about buffers a little

READ.ME: A BUFFERS SUBSTITUTE?

Lately, a new twist has appeared in buffer allocation. The auxiliary program BUFFERS.COM, supplied with Quarterdeck Office System's QEMM 5.0 and QRAM memory managers, can load buffers into high memory (memory above 640K), giving you the advantage of additional buffers without taking up valuable space in low memory. But you still must allocate a couple of buffers in CONFIG.SYS even though BUFFERS.COM is run from AUTOEXEC.BAT.

BUFFERS.COM doesn't always buy you much, however. Any system sophisticated enough to be using these memory-management products probably also uses a cache; so, the number of buffers required will be small and, therefore, so will the savings from loading buffers into high memory.

—J.D. & N.A.

differently. DOS buffers constitute a very crude cache; that is, they save data in (fast) memory to reduce (slow) disk accesses. If you have another caching program active, you'll have two programs performing the same function. This wastes processing time and, more important, wastes memory, because the same data is kept in two places: the DOS buffers and the cache.

With a cache active, you should set up only two or three DOS buffers to keep wasted memory to the absolute minimum. (If you go below two or three buffers, DOS can be very slow at the work it has to do before the cache is loaded.) But if your disk cache doesn't support floppy drives, use a higher BUFFERS setting to speed up floppy disk operations. Most cache manufacturers recommend this practice, because using disks without buffering or caching can create a real bottleneck. Of course, if your cache comes with a different recommendation, you should follow it.

COUNTRY=*nnn [ppp]*

COUNTRY is used to set the country code for DOS. It is supported in MS-DOS 2.1 and later and in PC-DOS 3.0 and later.

The country code determines the way in which DOS handles date, time, and currency symbols, providing a means of ensuring they meet national conventions. For example, in many European countries, the date is written as *dd.mm.yy*; in the United States, the standard form is *mm/dd/yy*; and in Japan, it is *yy/mm/dd*. The variable *nnn* is the international telephone dialing code for the country for which you want to set up your system, and *ppp* is the code page for that country. Dialing codes are listed in the appendix of many DOS manuals (and in most telephone books). The optional code-page number specifies the display and keyboard character set; these numbers also are listed in the appendixes of most DOS manuals.

Another optional parameter is the name of a country-information file, which generally is COUNTRY.SYS. This parameter may include a drive name and pathname.

If, for example, you want to set the country code for Germany, without changing the code page or country-information file, you can type the following command:

```
COUNTRY=049
```

If the COUNTRY command is not included in CONFIG.SYS, DOS uses the default U.S. values: country code 001, code page 437, and no country-information file.

DRIVPARM=*parameters*

The MS-DOS directive DRIVPARM never was implemented officially in PC-DOS. Its primary use is for specifying parameters for floppy drives other than those assumed by DOS. The main reason for using DRIVPARM in CONFIG.SYS is to permit more than two floppy drives or to set up a nonstandard drive. We will not discuss DRIVPARM in detail, because external device drivers, such as DOS 3.3's DRIVER.SYS, provide a better solution for accessing nonstandard floppy drives.

FCBS=*n,p*

FCBS tells DOS how to configure access to files using file control blocks (FCBs) when running on a network or with file-sharing support (SHARE.EXE) loaded. It is supported in DOS 3.0 and later.

File control blocks were introduced in DOS 1.0, but since the release of DOS 2.0, Microsoft has recommended that programs use the newer file-handle method for accessing files (see the information below on the FILES directive). Nevertheless, a number of programs, including DOS itself, continue to use the old FCBS file-access methods.

The *n* parameter sets the maximum number of files that may be open at any one time using FCBs; the default value is 4. The *p* parameter sets the number of these files that are protected against automatic closure by MS-DOS; the default value is zero. In general, the default settings are sufficient. If you receive an "FCB unavailable" error message, or something similar, when running a program, try increasing the value of *n* in the FCBS statement. The FCBS values are used only when SHARE.EXE or networking software is loaded and are meaningless in other situations.

To set the number of FCBs for use under SHARE or a network to eight, with four protected from automatic closure, type:

```
FCBS=8,4
```

FILES=*n*

FILES probably is the most commonly used parameter in CONFIG.SYS. It sets the maximum number of files that may be open at any time using file handles. (File handles are the newer and recommended method of opening files and the one used by most programs; the older method is file control blocks.) Under DOS 5.0, the number of file handles may range from eight to 255. The command for establishing 30 file handles is:

```
FILES=30
```

Under DOS 5.0, the default value for FILES is 30. Prior to DOS 5.0's release, the default value was 8. For most PC users, this is too few files. These days, even relatively simple programs may use more than eight files at a time, and it is common for programs to require a minimum FILES setting of 20. Some applications, especially database programs or those with built-in database capability, require many more. In virtually all cases, you should set FILES to 20 or more. Using DOS's old default setting of eight files often slows down the system and sometimes halts the program.

The effect of setting the number of files is to adjust the size of an internal table that DOS uses to track all open files. This table takes up about 50 bytes of memory per file;

READ.ME: System File Handles and Process File Handles

The FILES directive sets the number of file handles available to DOS, which controls the number of files that may be open at one time. File handles also are referred to as system file handles; consequently, the file-handle table sometimes is called the system file-handle table. This table contains one entry for every file that is open in your system.

When any program is started by DOS, it also gets its own table for "process" file handles. This table is stored in memory belonging to the program and exists only while that program is loaded. It is used by DOS to establish a correspondence between file handles used within the program and the entries in the system file-handle table. It contains 20 entries, and the only way to expand it is for the program to do so through its own methods or, in DOS 3.3 or later, through a DOS function call. The size of this table is not affected by the FILES setting in CONFIG.SYS.

If you temporarily exit, or shell, to DOS from inside an application or run a multitasker such as Quarterdeck Office System's DESQview, you may have several programs running at once. Each program in use has its own process file-handle table and can open as many as 20 files (more if the program expands its table). Sometimes DOS runs out of space in the system file-handle table before your program opens all its files. When this happens, you'll probably receive an "Out of file handles" error message or something similar. You can resolve this problem simply by increasing the FILES value in CONFIG.SYS.

In other circumstances, the reverse situation may arise: Your program may run out of file handles before DOS does. This condition also produces an "Out of file handles" error message, but adjusting the FILES setting in CONFIG.SYS won't set things right, because the program doesn't have enough file handles in its own internal table. The only solution in such cases is to check with the program's publisher to determine whether a way exists to tell the program to allocate more file handles internally.

Occasionally, a program may exhibit the behavior described in the preceding paragraph only when it is run from inside another program (say, from a menu). This occurs because the "parent" program isn't closing all of its files before starting your program. In this case, the files can be "inherited" by the "child" program, taking up unnecessary space in its process file-handle table and leading to "Out of file handles" errors. In such circumstances, your recourse is to contact the manufacturer, run the program directly from the DOS prompt, or get another menu program.

—J.D. & N.A.

so, for every 20 files specified, about 1K of additional memory is required for the file table. This gives you an incentive to keep the number reasonably low.

In most cases, you can find the appropriate value for FILES by examining the recommendations that come with the software you run. Use the highest value recommended by any of your software packages. If you receive an "Out of file handles" message, or something similar, try increasing the value of FILES to solve the problem. This solution does not always work, however, and you may have to consult your application's developer for assistance (see the sidebar above "READ.ME: System File Handles and Process File Handles").

As with BUFFERS, Quarterdeck Office System's QEMM and QRAM memory managers offer an alternative to making room for files in low memory. Invoking the FILES.COM program in AUTOEXEC.BAT lets you load part of the file-handle table in high memory (memory above 640K), saving low memory for applications and the like. To start DOS properly, you still must allocate at least five files in CONFIG.SYS, however. FILES.COM doesn't save a huge amount of memory, because each file-table entry takes only 50 bytes, but it can save you a couple of kilobytes or more if you use numerous files.

LASTDRIVE=*x*

LASTDRIVE, supported in DOS 3.0 and later, sets the highest disk drive letter DOS will recognize. The drive must be designated by a single letter, from *A* to *Z*. (Note that the colon character is omitted after the drive letter.)

If you load device drivers in CONFIG.SYS to set up additional drives (for example, using DRIVER.SYS to access a third floppy drive or invoking VDISK.SYS to set up a RAM disk), DOS allocates letters to those drives automatically. But if you allocate drive letters after CONFIG.SYS is processed (for example, with the DOS SUBST command or a RAM disk loaded in AUTOEXEC.BAT), you'll need to set LASTDRIVE to "make room" for the additional drive letters.

We recommend setting LASTDRIVE to the highest drive letter you normally use, plus one. This provides a little room to maneuver, without taking up space for unnecessary drive information. If you set LASTDRIVE to a value less than the highest drive letter already in use, DOS ignore the directive.

You can set the last drive letter to *M* by adding this line to your CONFIG.SYS file:

```
LASTDRIVE=M
```

STACKS=*n,s*

STACKS tells DOS how many stacks to allocate for interrupt processing. It is supported in DOS 3.2 and later.

Your PC's operation depends on

external signals from various pieces of hardware—disk drives or your system's timer chip, for example. These signals are called interrupts. They stop the normal flow of processing and temporarily transfer control of the microprocessor to a special program called an interrupt handler, which deals with whatever external condition caused the interrupt.

All programs in the PC use an area of memory called a stack to hold temporary data and keep track of where to go when a subprogram, such as one of the interrupt handlers, finishes its work. When many interrupts occur, quite a bit of space is needed to avoid running out of room on the stack. Because some programs lack sufficient room on their internal stacks to handle numerous interrupts, DOS maintains a pool of stacks, which it assigns for use by interrupt handlers. The STACKS directive in CONFIG.SYS sets the size of this pool.

The *n* parameter sets the number of stacks; its value may range from 8 to 64. The default value is 9. The *s* parameter sets the size of each stack in bytes. The range is from 32 to 512 bytes, and the default is 128. If you receive "Internal stack error" messages when your system is busy, try increasing the value of *n*.

The stacks allocated by this directive can take up quite a lot of memory. For example, if you use this command:

```
STACKS=32,256
```

you will use 8K of memory (32 x 256 bytes). In many cases, program stacks are big enough to accommodate the interrupt handlers in your system; so, you can save memory by disabling DOS stacks with the following command:

```
STACKS=0,0
```

[Note: On certain computers, the command `STACKS=0,0` causes problems. If your system locks up or behaves erratically or you get an insufficient stack space message, add stack space.]

Device Drivers

A device driver is simply a program that lets DOS communicate with a particular device, such as a network, mouse, memory board, or RAM disk. You load device drivers into memory when the system starts by placing DEVICE directives in CONFIG.SYS. They remain in memory until you change CONFIG.SYS and the system is rebooted.

Device drivers were introduced in DOS 2.0 to allow manufacturers to write their own software for add-on products for the PC and to make DOS more flexible. As PC software has matured, software developers have begun using device drivers for purposes other than controlling physical devices. For example, two popular 386 memory managers, Qualitas's 386 Max and Quarterdeck's QEMM both are implemented as device drivers. Yet they control a "device"—memory—that is integral to the system.

The syntax for a CONFIG.SYS DEVICE directive is:

```
DEVICE=filename [parameters]
```

where *filename* is the name of the file containing the device driver and may include the drive letter and pathname. (The default drive and path is the root directory of the boot disk.) Most drivers have a SYS extension. The optional parameters specify any additional information needed by the driver; its format varies and can be found in the documentation for the driver.

We recommend that you store device drivers somewhere other than in the root directory of your boot disk. A careless command issued while in the root directory can destroy a crucial driver file inadvertently, and you won't notice the problem until the next time you boot up. Because DOS lets you specify a path in the DEVICE statement, you can keep driver files with their respective software or store them in a directory all their own.

DOS comes with a number of device drivers, which you may install if you want to take advantage of their capabilities. The following standard drivers are included with most versions of DOS.

ANSI.SYS

ANSI.SYS provides improved screen and color control. Many different options are available; see your DOS manual for assistance with this driver.

DRIVER.SYS

DRIVER.SYS provides control for external floppy disk drives and additional disk formats for existing floppy drives. DRIVER.SYS is distributed with DOS 3.3 and later; it also was available in some versions of DOS 3.2. The parameters for DRIVER.SYS are:

- /D:*n*—indicates the drive number and must be the first one on the line. Substitute the appropriate value for *n*, using zero for drive A, 1 for drive B, and so on.
- /C—indicates that the disk drive can tell the system whether you have opened the door and removed the disk. (The system assumes that the disk has not changed until it gets a "change line" signal.)
- /F:*n*—tells the size and capacity (form factor) of the drive. The options are as follows (the default value is 2):

```
0 = 320K or 360K, 5.25-inch
1 = 1.2MB, 5.25-inch
2 = 720K, 3.5-inch
3 = 8-inch single-density
4 = 8-inch double-density
5 = hard disk
6 = tape drive
7 = 1.44MB, 3.5-inch, or other
```

- /H:*n*—indicates the number of heads. You supply a value for *n*.
- /N—indicates that the disk is not removable.
- /S:*n*—indicates the number of sectors per track. You supply a value for *n*.

- /T:*n*—indicates the number of tracks on each side of the disk. You must supply a value for *n*.

You can load DRIVER.SYS to control an external floppy drive or to provide better control over an existing drive, especially when you have an older PC whose BIOS does not recognize the capabilities of a newer drive.

For example, suppose you have a 1.44MB, 3.5-inch floppy drive installed as drive B, but your system's BIOS handles only older 720K, 3.5-inch drives. If you have a hard disk partitioned into two logical drives (C and D), you can install DRIVER.SYS to create a new drive, E (the next available drive letter), which permits you to access the full capacity of your 1.44MB drive. The command to do this is:

```
DEVICE=C:DRIVER.SYS /D:1 /C /F:7
```

Installing DRIVER.SYS in this way, makes the drive accessible as a 720K drive (drive B) and as a 1.44MB drive (drive E).

Because 1.44MB drives can read from and write to 720K disks, you can refer to the drive as drive E for all operations. The DOS FORMAT command would automatically format the drive's disks to hold to 1.44MB. To specify a 720K format, you can use FORMAT's command-line options or simply type:

```
FORMAT B:
```

TIP

LET YOUR FINGERS DO THE EDITING

DOS 5.0's editor, Edit, offers both mouse compatibility and cut-and-paste features. But when you're typing at speed, you may not always want to reach for the mouse to select a word or a line. Fortunately, the Editor also provides a full range of keyboard shortcuts. Here are several of the most useful keyboard tricks:

- Press Home to move to the beginning of a line; End takes you to the end of a line.
- Press Ctrl-left arrow to move one word to the left and Ctrl-right arrow to move one word to the right.
- Press Ctrl-T to delete the word at the cursor. (This works only if the cursor is on the first character of the word.)
- Press Shift-down arrow to select the current line and the line below it.
- Press Shift-up arrow to select the current line and the line above it.
- Press Shift-Ctrl-right arrow or Shift-Ctrl-left arrow to select one word at a time. You can hold down Shift and Ctrl and keep pressing the right-arrow key to move right as you select words; pressing the left arrow instead moves you left.
- Press Ctrl-Y to cut the current line.
- Press Ctrl-Q and type *Y* to cut the information from the cursor to the end of the line.
- Press Ctrl-Ins to copy the information you've selected.
- Press Shift-Del to cut what you've selected.
- Press Shift-Ins to paste what you've cut or copied.

An easy way to remember the last two shortcuts is to remember that cutting and pasting shifts text from one location to another. Shift-Del deletes text from your document and deposits it in Edit's buffer; Shift-Ins removes information from the buffer, inserting it at the designated location in your document.

—Kay Yarborough Nelson

VDISK.SYS

VDISK.SYS, which is included with IBM's PC-DOS, lets you create a RAM disk. Other versions of DOS may include RAMDRIVE.SYS. [Note: RAMDRIVE.SYS uses a different syntax.]

A RAM disk is an area of memory that DOS and your programs can use as a disk drive, a very fast disk drive. It can provide a convenient place to store frequently used programs or temporary files. A wide variety of RAM disk drivers are available; VDISK.SYS is only one of them. Many others are more sophisticated and provide greater flexibility and control, but if you simply need a basic RAM disk in main or extended memory, VDISK.SYS should do the trick.

The program's basic command syntax (assuming VDISK.SYS is in your root directory) is:

```
DEVICE=C:VDISK.SYS disksize
  secsize dirsize /E:n
```

where *disksize* specifies the size of the RAM disk in kilobytes. You may allocate as little as 1K or as much memory as is available; the default is 64K. If the RAM disk is in main memory, VDISK.SYS always leaves at least 64K free after the disk is created, regardless of the size you specify.

Next, *secsize* is the sector size in bytes. The allowable values are 128, 256, and 512. Normally, you should set the sector size to 512 bytes (the PC standard), rather than the default of 128 bytes.

Next, *dirsize* is the number of entries (files) that may be stored in the RAM disk's root directory. The range is two files to 512 files. The default value of 64 should be sufficient, unless you place a large number of files in the root directory.

The optional parameters /E or /X tell VDISK.SYS to use a RAM disk in extended or expanded memory. In most cases, a RAM disk won't be large enough to be useful if you have to squeeze it into main memory; so, you'll probably use this parameter whenever you have occasion to use VDISK.SYS.

You may follow /E with a number (/E:*n*) to limit the number of sectors VDISK.SYS transfers to extended memory at one time. This capability is available only in PC-DOS 3.1 and later. The value of *n* must fall between 1 and 8; the default setting is 8. You should reduce the value of *n* to less than 8 only if you have trouble with I/O devices while using VDISK.SYS—if, for example, characters are lost by a printer or communications port. Reducing *n* limits the amount of time VDISK.SYS disables processing of interrupts from these devices while it is transferring data to and from extended memory.

Peaceful Coexistence

In addition to these standard drivers, your system may use others. In some cases, you will need drivers because your hardware—perhaps a network interface or mouse—requires them. In other cases, drivers are used to provide utility functions, such as memory management.

One tip for solving compatibility problems with device drivers is to try rearranging the DEVICE statements to load the drivers in a different order. Sometimes, drivers that appear to conflict with each other will get along fine when a different load order is used. Unfortunately, the number of possible combinations of drivers is so large that the only tip we can give you is to experiment and see what works. It isn't hard to do as long as you are systematic and thorough and you change the order of only two drivers at a time.

If you do have compatibility problems (drivers that don't work or don't work correctly), it's often best to remove all the drivers you can and reload them one by one until the problem recurs. This approach often can identify the culprit faster than random rearrangements or guesswork.

Recent software improvements and DOS 5.0 have added a new capability for device drivers: the option of "loading them high" by placing them in the area between 640K and 1MB and thus conserving low memory for other uses. The major programs that provide this capability for versions of DOS prior to 5.0 are Qualitas's 386 Max (for 386s) and MOVE'EM (for other systems) and Quarterdeck's QEMM (for 386s) and QRAM (for other systems).

Loading drivers high with Quarterdeck's QEMM requires a line such as the following:

```
DEVICE=D:LOADHI.SYS filename
  parameters
```

In this case, as far as DOS can tell, the driver being loaded is LOADHI.SYS. But this command actually loads the real driver (whose name is specified in *filename* and whose parameters are listed in *parameters*) and stores it in high memory.

Loading drivers into high memory is a bit of an art. Some drivers require considerably more memory to load than they do to run, and some won't work at all if loaded high. Couple that with the need to load drivers in the right order and the division of a typical system's "high memory" into two or more pieces of noncontiguous memory space, and you have the makings of a really tricky procedure.

The Command Processor

The one remaining directive you can use in CONFIG.SYS is SHELL. SHELL specifies the name of the command processor to be loaded when DOS is started and lists any parameters for the command processor. If you don't specify a command processor in CONFIG.SYS, DOS looks for its default command processor, COMMAND.COM, searching for it in the root directory of the boot drive.

The syntax for SHELL parallels that for DEVICE:

```
SHELL=filename parameters
```

where *filename* is the name of the file containing the command processor, including drive and pathname, if necessary, and *parameters* specifies any additional information needed by the command processor.

If you don't store COMMAND.COM in your boot disk's root directory and you want to simultaneously invoke COMMAND.COM as the command processor and set the environment, you can use SHELL to do this. (As with device drivers, placing COMMAND.COM in a directory other than the root directory affords a measure of protection against accidental deletion or damage.) In this case the syntax is:

```
SHELL=[d:][pathname].COM /E:n /P
```

where *[d:][path]* provides the drive and pathname for COMMAND.COM, *n* is the size of the DOS environment in bytes (from 160 to 32,768 in DOS 3.2 and later) or in 16-byte paragraphs (DOS 3.1), and /P tells DOS that this is the primary command processor and that AUTOEXEC.BAT should be executed. If the /E:*n* switch is omitted, a default of 160 bytes, or 10 paragraphs, is assumed. If the /P switch is missing, AUTOEXEC.BAT is not executed.

The ability to select an alternate command processor was introduced in DOS 2.0. But until DOS 3.0 was released, COMMAND.COM included some undocumented—but necessary—functions that programmers writing alternative command processors could not easily duplicate. As a result, while it was theoretically possible to completely replace COMMAND.COM under DOS 2.0, it is practical only under DOS 3.0 and later. Alternate command processors generally can run under DOS 2.0, but you must load them after COM-

MAND.COM rather than as a replacement for it. If you have an older version of DOS and want to use a replacement command processor, upgrading to DOS 5.0 would be well worth your while.

A program that replaces COMMAND.COM must provide certain features required by all programs running under DOS, such as critical error and Ctrl-Break handling. Beyond these basic features, the replacement command processor can implement any set of commands the author chooses; the resulting system need not look like DOS at all. In fact, until recently, the primary programs that took advantage of the ability to install an alternate command processor were shells (another term for command processors) designed to make DOS look and work like Unix.

Recently, many PC users have begun using a shareware replacement for COMMAND.COM, J.P. Software's 4DOS. 4DOS provides the same basic commands as COMMAND.COM, plus a variety of new commands and enhancements. The following example shows how you can use SHELL to load this particular alternate command processor:

```
SHELL=C:4DOS.COM /E:2048
/A:2048 /U /P
```

TIP PUT MORE LINES ON YOUR SCREEN

If you'd like to see more than the typical 25 lines of text on your screen, perhaps DOS can help. Many monitors and video adapters let you use the DOS MODE command to switch to a 43- or 50-line display. To find out whether your video components permit this, first be sure your CONFIG.SYS file contains a statement that loads the ANSI.SYS device driver. On most systems, ANSI.SYS resides in the C:\DOS directory, so the correct statement in CONFIG.SYS is:

```
DEVICE=C:\DOS\ANSI.SYS
```

(If ANSI.SYS is stored in a directory other than your C:\DOS directory, substitute the correct pathname.)

To add the statement to CONFIG.SYS, use Edit, Edlin, or any text editor or word processor that lets you save files in text (ASCII) format. After adding the ANSI.SYS line and saving the new version of CONFIG.SYS, restart your computer to load ANSI.SYS. Then, at the DOS prompt, type:

```
MODE CON: LINES=43
```

or

```
MODE CON: LINES=50
```

If you prefer a 43- or 50-line display to the normal 25-line display and want your computer to start up in that mode, you can add one of the above MODE statements to your AUTOEXEC.BAT file.

Not all display adapters support the 43- and 50-line settings. Try the commands from the DOS prompt before adding one of them to your AUTOEXEC.BAT file.

—Jeff DeTray

One special note regarding the SHELL directive: A bug in all versions of DOS from 2.0 through 4.01 limited the length of the parameters portion of the line. The length of the entire line is immaterial; it is only the parameters portion that matters. If the parameters (including any leading and trailing spaces) take more than 32 bytes, the system will not boot properly. Symptoms include garbage in the DOS environment and perhaps incorrect behavior on the part of the command processor. This bug is not a problem with COMMAND.COM, because the longest useful parameter string is /E:*nnnnn* /P, which is less than 32 characters. However, it may easily affect replacement command processors, which must provide a mechanism for dealing with this shortcoming.

For those interested, the bug is that DOS has only a 29-byte buffer for the parameters portion of the SHELL directive, and it does not check whether the parameters exceed this length. The first 3 bytes beyond the buffer can be overwritten without causing trouble; hence the 32-byte limit. The limit may be slightly longer than 32 bytes in some versions of DOS.

CONFIG.SYS Examples

This section provides three sample CONFIG.SYS files, which we've labeled "beginner," "intermediate," and "advanced." The terms refer not to the level of knowledge required to use the systems for which the files were designed, but to the level of familiarity with CONFIG.SYS directives and the system itself needed to create each file.

Beginner

This version of CONFIG.SYS is pretty simple. Our hypothetical user needs to expand the FILES and BUFFERS settings; load a single device driver, ANSI.SYS; and—per our earlier recommendation—invoke COMMAND.COM from the DOS directory rather than from the root directory. These are the four lines to accomplish this:

```
FILES=20
BUFFERS=16
DEVICE=C:.SYS
SHELL=C:.COM /P
```

Intermediate

This user has a slightly more complex setup. The system has an expanded memory board, which requires an EMS driver, and a mouse, which requires a mouse driver. The FILES directive allows a lot of files to be open at once, but a cache is in use; so, the number of buffers is much smaller than the number of files.

We assume this user is running DOS 3.3; so, the environment size is stated in bytes—1,024 (1K) of them, far larger than DOS's default setting of 160 bytes. The user also has a large hard disk partitioned into two drives. Most of the system's software resides on drive D. Drive F is defined as an alternative drive letter for the second disk drive, a 3.5-inch floppy, which allows it to function as a 1.44MB drive.

Ten lines compose this CONFIG.SYS version:

```
FILES=64
BUFFERS=3
STACKS=0,0
DEVICE=D:.SYS
DEVICE=D:.EXE
DEVICE=D:.SYS
DEVICE=D:.SYS
DEVICE=D:.SYS /D:1 /C /F:7
DEVICE=C:.SYS 384 512 64 /E
SHELL=D:.COM /E:1024 /P
```

Advanced

This user has a 486 system with a number of device drivers, most of which are loaded high. The system uses 4DOS, the replacement for DOS's COMMAND.COM command processor mentioned earlier. Because this user has employed special programs—both shareware and commercial—to improve hard disk speed, handle memory, and add a margin of safety to the computer, the configuration file is complex. For this reason, we organized this file in tabular format, explaining each line or group of lines separately (see "Advanced CONFIG.SYS File" above).

We hope the preceding advice and our example CONFIG.SYS files help you feel more comfortable with your start-up environment. If you take advantage of some of the items mentioned in this article, you should see a significant boost in your PC's performance. ■

John Dvorak, a widely syndicated columnist, is author of PC Magazine's *"Inside Track" column and also writes monthly columns for* PC Computing *and* MacUser *magazines. He and coauthor Nick Anis have previously collaborated on* Dvorak's Guide to DOS & PC Performance *and* Dvorak's Guide to PC Telecommunications, *both published by Osborne/McGraw-Hill.*

ADVANCED CONFIG.SYS FILE

```
LASTDRIVE=J
BUFFERS=2
FILES=8
```

This sequence sets up the last drive as J and tells DOS to provide two buffers and eight files. (More buffers and files are made available later by invoking QEMM's BUFFERS.COM and FILES.COM in AUTOEXEC.BAT.)

```
DEVICE=E:\QEMM\QEMM.SYS RAM=B000-B7FF RAM=CC00-DFFF
```

Loads Quarterdeck's QEMM memory manager, specifying two areas of high memory to use for loading drivers and TSRs high.

```
DEVICE=E:\QEMM\LOADHI.SYS C:\XDRIVE.SYS /D:2 /F:1 /U
```

Loads a floppy disk driver similar to DRIVER.SYS into high memory.

```
DEVICE=E:\QEMM\LOADHI.SYS C\KEYSTACK.SYS
```

Loads the 4DOS KEYSTACK.SYS driver into high memory.

```
DEVICE=E:\QEMM\LOADHI.SYS E:\PCKWIK\PCKRAMD.SYS /S:512
/d:128
```

Loads Multisoft's Super PC-KWIK RAM disk program into high memory, setting the sector size to 512 bytes and the number of directory entries to 128.

```
DEVICE=E:\QEMM\LOADHI.SYS E:\UTIL\BURNDEV.SYS
```

Loads into high memory Chris Dunford's freeware screen blanker, BURNDEV, which blanks the screen to avoid the "burning in" of an image when the PC is inactive.

```
DEVICE=E:\PKII\KYBD.BUF /B64/n1
```

Loads the shareware PowerKit II keyboard-buffer program.

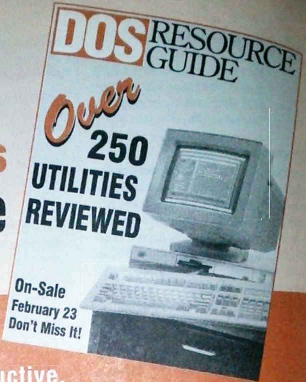
```
.DEVICE=F:\PERI\SYSLOAD.SYS /P=F:\PERI\PS.COM
```

Loads the Periscope debugger, which this user wants to happen only occasionally. Starting the line with a period (.) ensures that it is not executed but keeps the line in place so that you don't have to recreate it later. DOS issues a harmless error message when it executes this line.

```
SHELL=D:\4DOS\4DOS.COM /L:D:\4DOS /E:U /A:5120U /U /P
```

Loads 4DOS as the command processor, specifies a 5K alias buffer, and loads the master environment and the command processor itself into high memory.

Coming In February!



DOS RESOURCE GUIDE

Over 250 UTILITIES REVIEWED

On-Sale
February 23
Don't Miss It!

DOS RESOURCE GUIDE's Special Utilities Issue

A complete reference to hundreds of software tools that make life at the C:> prompt more fun and productive.

IN THIS SPECIAL ISSUE

YOU'LL LEARN HOW UTILITIES CAN:

- Improve System Performance
- Manage Files and Directories
- Troubleshoot PC Hardware
- Eradicate Viruses and Hardware Glitches
- Recover Lost Data

YOU GET DETAILED DESCRIPTIONS AND BUYING INFORMATION ON MORE THAN 250 INDISPENSABLE DOS UTILITIES:

- Memory Managers
- DOS Shells
- Disk Optimizers
- Diagnostic Programs
- Backup Software
- File Compression Utilities
- **Much, Much More!!!**

SPECIAL FEATURES:

- How to Choose the Right Utility
- Where to Find Shareware Alternatives

PLUS: Dan Gookin's Six Must-Have Utilities

FREE BONUS UTILITY!

This Special Issue contains the complete QuickBasic listing for Portal, a handy file transfer program for moving data from one PC to another.

DOS Resource Guide's Special Utilities Issue is not available by subscription. Look for it only on newsstands and in bookstores—wherever computer magazines are sold.

On-Sale February 23 . . . Don't Miss It!

Greetings From Your DOS Prompt

Why settle for DOS's default black-and-white C:> message when the operating system offers plenty of opportunity for personalizing your prompt and color scheme?

by Antonio and Alfonso De Pasquale

Most PC users don't give much thought to the DOS prompt. They're satisfied if it tells them the current drive and directory. But you can customize your prompt to display other information, such as the date, time, and your DOS version number. You also can use the prompt to change text and background colors.

You change the prompt by issuing the PROMPT command, followed by a text string, parameters, or both. (See the table "Prompt Parameters" for a list of the parameters available to all versions of MS-DOS.)

You can type your command at the DOS prompt, or you can include it in your AUTOEXEC.BAT file so that the command executes automatically when you turn on your computer. The syntax is as follows:

```
PROMPT [text]
```

To see how it works, type in the following command, and press the Enter key:

```
PROMPT This is my computer
```

Your DOS prompt will look like this:

```
This is my computer
```

This new prompt remains in effect until you change it with another command or until you shut off your PC.

PROMPT: One of the more entertaining as well as helpful of the DOS commands. The PROMPT command controls how the prompt (C:>) stares out at you from the display.

COMMAND TYPE: Internal

AVAILABLE: MS-DOS 2.0 and later

In addition to text, you can display system information, such as the date, time, drive letter, or directory name. This is where parameters come in. For example, the parameter $N gives the current drive, and $G provides the greater-than (>) sign. Thus, the command:

```
PROMPT $N$GThis is my computer
```

yields the following:

```
C:>This is my computer
```

The most common parameters are $P$G, which, when combined, display the drive, directory, and greater-than sign. Assuming you are in your DOS subdirectory, this command:

```
PROMPT $P$G
```

displays the following prompt:

```
C:\DOS>
```

Here's another example. This one gives the time, rounded to the nearest minute:

```
PROMPT The time is $T$H$H$H$H$H$H
```

The result will look something like this:

```
The time is 10:36
```

Each $H parameter backspaces one character. Otherwise, the time shows the hours, minutes, seconds, and hundredths of a second, as in:

```
The time is 10:36:24.02
```

You also can combine numerous parameters within one PROMPT statement. For example, the following prompt displays the date and time on separate lines:

```
PROMPT $D$_$T
```

The result might be:

```
Fri 08-21-1992
15:47:11.13
```

In this example, $D causes the date to be displayed, $_ issues a line feed, and $T displays the time. Taking this example further, you also can add the names of the current drive and directory. Typing:

```
PROMPT $D$_$T$_$P$G
```

gives you a display such as:

```
Fri 08-21-1992
15:49:30.75
C:\XY\DRG>
```

Here, $P displays the current drive and directory, and $G provides the greater-than symbol. You can add descriptive text to the previous example, as follows:

```
PROMPT Date: $D$_Time:
  $T$_Directory: $P$G
```

This command gives a prompt that looks like this:

```
Date: Fri 08-21-1992
Time: 15:50:48.36
Directory: C:\XY\DRG>
```

You also can add company information to your prompt, a technique used by network administrators. As an example, the following prompt will display *IBM* - and the current drive and directory:

```
PROMPT IBM - $P$G
```

PROMPT PARAMETERS

CODE	FUNCTION
$T	Displays the current time
$D	Displays the current date
$P	Displays the working directory of the default drive
$V	Displays the MS-DOS version number
$N	Displays the default drive
$G	Displays the > character
$L	Displays the < character
$_	Enters a carriage return/line feed
$E	Displays an ASCII escape (code 27)
$Q	Displays the = character
$B	Displays the | character
$H	Backspaces one space
$$	Displays the $ character

Adding Color to the Prompt

In addition, you can use color to give your prompt extra impact. First, you must load ANSI.SYS or a compatible device driver, such as FANSI.SYS or SIZZLE.SYS. (Such drivers may be found on bulletin boards or often are supplied by the vendor who sold you your computer.) With a text editor, add the line:

```
DEVICE=ANSI.SYS
```

to your CONFIG.SYS file. Then, reboot your computer.

SCREEN COLORS

FOREGROUND COLORS		BACKGROUND COLORS	
Number	**Color**	**Number**	**Color**
30	Black	40	Black
31	Red	41	Red
32	Green	42	Green
33	Yellow	43	Yellow
34	Blue	44	Blue
35	Magenta	45	Magenta
36	Cyan	46	Cyan
37	White	47	White

You add color with an escape sequence, which begins with the characters $E[and then provides a number for the foreground color and one for the background color. The following prompt command displays the drive and directory in blue text:

```
PROMPT $E[34;40m$P$G$E[37;40m
```

(Note that codes such as $E, $P, $G can appear in upper- or lowercase, but *m*, the function used to set colors and text attributes, must be in lowercase.) The string `$E[34;40m` changes the screen colors to blue text on black background, which makes the information that immediately follows it—$P$G, the drive, directory, and greater-than sign—appear in blue text on a black background. The second sequence, `$E[37;40m`, resets the colors to white on black so that the characters you type at the prompt appear in white against a black background.

The table "Screen Colors" lists available colors and their codes. Here's a more advanced example that displays the date, time, drive, and directory in green text on a black background:

```
PROMPT $E[32;40mDate: $D Time:
  $T$_$P$G$E[37;40m
```

A Challenge to Our Readers

You also can use the PROMPT command to position the cursor, reverse screen type, and make portions of screen text blink. Can you figure out how to accomplish these tasks? The information you need should be in your DOS manual. (Here's a hint: Look for information on ANSI escape sequences.) ■

Free-lance writers Antonio and Alfonso De Pasquale live in East Boston. They are frequent contributors to DOS Resource Guide.

The Fairest DOS Command of All

It's time to look more closely into MIRROR, the final safeguard against hard drive disaster.

by Dan Gookin

Once upon a time, when DOS users mistakenly deleted a file or—perish the thought—reformatted a hard disk, they despaired of ever seeing their data again. Using a PC was scary back then! Fortunately, computing isn't so frightening any more.

Nowadays, DOS 5.0's UNDELETE and UNFORMAT commands provide instant relief for the minor file mishaps that befall most users. Type *UNDELETE*, followed by the name of a newly departed file, answer a few prompts on the screen, and the file lives again. The amazing UNFORMAT command works similar wonders with accidentally zapped disks. Yet although you already may know about UNDELETE and UNFORMAT, you may not realize that they both can work even more smoothly if you take advantage of DOS 5.0's often overlooked MIRROR command. (The table "Using MIRROR, UNDELETE, and UNFORMAT" summarizes the commands' uses.)

MIRROR offers double protection against the threat of data loss. It's that reserve parachute strapped to your PC's belly "just in case." The problem is, because DOS often can figure out by itself how to undelete a file or unformat a hard drive, few people bother with MIRROR.

Maybe it's the command's name that prompts users to ignore MIRROR and assume it has something to do with the monitor—or perhaps the PC's vanity. Or maybe it's MIRROR's ability to handle three different, though related, duties: recording a hard drive's partition table, saving information about vital areas of a disk, and monitoring deleted files with deletion tracking.

The name *mirror* implies a duplicate image. That's MIRROR's primary job: creating a copy of critical areas on your disk, including your hard drive's file allocation table (FAT); root directory; and boot sector, the portion of your disk responsible for starting DOS on your computer. MIRROR also can save an extra copy of a hard drive's partition table, the area describing how DOS uses the drive. Damage to any of these sensitive areas can mean the loss of important files—or even everything—on your hard drive. MIRROR can't prevent damage from occurring, but it does make recovery a much more pleasant experience.

MIRROR: DOS's first try at data protection creates two files on a specified drive. The first, MIRROR.FIL, saves a copy of a disk's FAT and root directory. Use it with DOS's UNFORMAT to restore an accidentally reformatted disk. PCTRACKR.DEL, a hidden file that lists your deleted files, can be used with UNDELETE to restore a lost file.

Program Type: External

Available: MS-DOS 5.0

Disaster-proofing Your PC

To help safeguard the information on your hard drive, you need to do two things. First, create an emergency boot disk so that you can get your system up and running in the event that your hard drive fails (see the sidebar "Emergency Preparedness"). Second, get yourself to the DOS prompt without delay and type the following variation on the MIRROR command:

```
MIRROR /PARTN
```

The /PARTN switch tells MIRROR to create a file named PARTNSAV.FIL and store in it information about all the partition tables for all of your hard drives. Onscreen instructions direct you to insert the floppy disk that you want to hold PARTNSAV.FIL. I recommend putting it on your PC's emergency boot disk.

You probably will need to use the MIRROR command with the /PARTN switch only once. Unlike everything else on an active hard drive, the partition table is modified

EMERGENCY PREPAREDNESS

In a perfect world, your hard disk would always be ready to do business when you needed it. But life isn't perfect, and neither are hard disks. To protect yourself in the event a disk disaster prevents you from accessing your hard drive, you need to keep an emergency bootable floppy disk handy. If you have multiple computers that you configure differently, you should create a customized emergency boot disk for each of them.

To get started, find a blank disk that fits in the floppy drive your system uses as a boot drive. If the disk you select is unformatted, format it by typing the following command at the DOS prompt:

```
FORMAT /S
```

This adds COMMAND.COM and the indispensable system files IO.SYS and MSDOS.SYS to the disk during formatting. You can make a previously formatted disk bootable by putting the disk in your boot floppy drive, logging onto that drive, and invoking the SYS command.

You may need more than just COMMAND.COM, IO.SYS, and MSDOS.SYS to put your system back in working order, however. Adding the following DOS utilities to your emergency boot disk will save you the trouble of searching for them on your backup DOS disk:

- CHKDSK.EXE, a utility that lets you check the integrity of a disk and fix problems you find
- DEBUG.EXE, a program that lets you read and modify memory, files, and disk sectors
- EDLIN.EXE, to patch AUTOEXEC.BAT or CONFIG.SYS (Preferable to DOS Edit, because it takes only 12K. The Edit command requires QBasic be loaded.)
- FDISK.EXE, DOS 5.0's disk partitioning software
- FORMAT.COM, DOS's disk formatting utility
- MEM.EXE, a utility that provides information on how your computer's memory is set up
- RESTORE.EXE, a program that lets you restore your file and directory structure—provided you used BACKUP.EXE to back up those files and directories
- SYS.COM, a utility needed to make a disk bootable
- UNDELETE.EXE, a program that permits you to resurrect deleted files
- UNFORMAT.COM, a utility for recovering files and directories when you accidentally reformat a hard disk

You also should consider including third-party utilities, such as an antivirus program and a backup program (with its attendant backup files), and the MIRROR command's PARTNSAV.FIL file. In addition, include any device drivers that are absolutely necessary to starting your system. SCSI and Bernoulli drivers fall into this category, as do drivers for disk compression packages such as Stacker. Disk compression programs store information in a special format, which you cannot read unless you have the proper driver. After singling out the drivers you need, create a CONFIG.SYS file for your boot disk, being careful to invoke the drivers in their proper order.

Among the device drivers you should omit are nonessentials, such as ANSI.SYS, Doskey, and memory-resident programs. It's also best not to include MIRROR. If you run the MIRROR command on a damaged hard drive, it may overwrite vital information kept in the MIRROR.FIL file—information you need for recovery.

If the list of necessary utilities grows too long, and you know you won't be able to fit all of them on one floppy disk, set aside another disk to hold the overflow. (You needn't make that disk bootable.) A backup program and its backup files are good candidates for placement on a separate disk.

Label your emergency disk appropriately and keep it near your PC. At least once a month, reset your computer with the emergency boot disk in place to ensure it still works. You also might want to review the list of utilities you've included to make sure you have everything you need to recover from a disk disaster. Finally, if you use a utility package that comes with its own emergency disks, consider storing them in the same place as your emergency boot disk. That way, you'll know where to look for the disks when you need help.

—D.G.

only when you add or remove a hard drive, or when you repartition a drive using the FDISK command. Once you've created and saved PARTNSAV.FIL, you just cross your fingers and hope you never need it. If anything does happen to your hard drive's partition table, you can recover using UNFORMAT and an extra copy of PARTNSAV.FIL you've stashed away. (I'll explain how to do this in a moment.)

With those precautionary measures out of the way, your next step is to start using MIRROR on a daily basis—by adding a command such as the following to your PC's AUTOEXEC.BAT file:

```
MIRROR C: D: E:
```

MIRROR creates a file named MIRROR.FIL in the root directory of each of the drives you mention. MIRROR.FIL contains a copy of the hard drive's boot sector, FAT, and root directory information—in case the "big scary" happens and one of these areas becomes damaged.

Every time you add a new file, delete an old one, or change the length of an existing file, DOS updates your hard drive's FAT. This means, of course, that to have MIRROR.FIL reflect the precise state of the FAT, you'd need to be constantly invoking MIRROR. Even the root directory often changes during a computing session. But don't go overboard: Running MIRROR once a day usually provides a sufficient margin of safety.

Incidentally, you don't need to use MIRROR on floppy disks. The FORMAT command automatically saves "unformat information" on each disk, unless you format a floppy using FORMAT's /U option to "unconditionally" format the disk. When you use /U, you can recover the files the disk used to contain. (The same restriction applies to hard drives—so be careful with the /U option!)

On the Right Track

Your drives' vital statistics are now in protective custody, but you still haven't done anything for your files. To put MIRROR to work safeguarding *them* from accidental deletion, you need to turn on the command's deletion-tracking feature by specifying a /T switch for each of the drives you want to track. The /T switch directs MIRROR to go "undercover" and run in memory-resident mode, where it can monitor the files that DOS deletes. The following command, for example, saves vital information on drives C and D and keeps track of any files deleted from those drives:

```
MIRROR C: D: /TC /TD
```

Note that in the second part of the command the *C* and *D* immediately follow the /T switch (you do not insert a colon) and that one /T switch is required for each drive.

Whenever you delete a file, or a whole swath of files, MIRROR "remembers" the names of those files and their locations on disk. When you try to recover a file using UNDELETE, DOS checks to see if deletion tracking is active. If it is, the files are recovered almost instantaneously, providing you specify the proper options for UNDELETE.

But deletion tracking has its downside. First of all, to use it you must make MIRROR memory resident, which means it takes up space that you may need for other things. Although you can load MIRROR "high" (this happens automatically if you use DOS's memory manager, EMM386.EXE), the program still takes up RAM and may conflict with other terminate-and-stay-resident programs.

Another drawback is PCTRACKR.DEL, the invisible file that MIRROR uses to keep track of the files you delete. On some hard drives, the default setting for the number of deleted files monitored by PCTRACKR.DEL permits the file to occupy as much as 55K of disk space (see the table "Default Settings for PCTRACKR.DEL"). You can, however, limit the number of files PCTRACKR.DEL tracks. The following command, for instance, tells MIRROR to monitor only the 100 most recently deleted files on drive C:

```
MIRROR C: /TC-100
```

Tracking those 100 files consumes about 18K of disk space; tracking 50 files requires only about 9K of disk space.

If disk space usually isn't at a premium, you'll seldom need to limit the number of permissible entries. On the other hand, if you're doing a lot of disk maintenance on particularly large directories, you may want to specify a huge value, say 500, to make sure deletion tracking will help you recover a hoard of files deleted accidentally. MIRROR can track a maximum of 999 files; 1 is the minimum entry value.

DEFAULT SETTINGS FOR PCTRACKR.DEL

Disk size	Default number of entries	Size of PCTRACKR.DEL
360K	25	5K
720K	50	9K
1.2MB or 1.44 MB	75	14K
20MB	101	18K
32MB	202	36K
32MB+	303	55K

Oops!

If you immediately realize that you've deleted a file by mistake, getting it back is no big deal. You simply type the UNDELETE command, following it with the name of the file you want to recover, as in this command:

```
UNDELETE MYDOC.TXT
```

UNDELETE searches out that file and lets you know whether it thinks it can recover the entire file. If deletion tracking is active, and information on the file you want to recover is stored in PCTRACKR.DEL, UNDELETE uses that information to quickly recover the deleted file. Without deletion-tracking information, recovery takes longer, and you must answer more prompts.

If you need to recover a group of files, the UNDELETE command can employ wildcards in its syntax:

```
UNDELETE filespec /ALL
```

where *filespec* is a wildcard specification, such as C:\SPREAD\MARCH\*.WKS, that indicates which files you want to recover. The /ALL switch directs UNDELETE to recover all files matching the wildcard. If deletion tracking is active, UNDELETE does not prompt you to confirm each file before recovering it.

You may use the /ALL switch even if you haven't activated deletion tracking, but the files must be renamed during recovery, because DOS deletes a file by erasing its initial character (see the sidebar "How Undeleting and Unformatting Work"). During recovery, UNDELETE supplies a pound sign (#)—or another character that makes the name of the recovered file unique—for the filename's missing first character.

Disaster Averted

If the mere thought of losing the initial character of a few files makes you jittery, the prospect of being unable to access your hard drive or accidentally reformatting a 120MB hard drive undoubtedly will severely elevate your blood pressure. Not to worry. If you take my advice about creating an emergency disk and pressing MIRROR into service, UNFORMAT frequently can solve partitioning and formatting problems in short order.

If DOS tells you that the partition table or boot sector for your boot hard drive is damaged, hunt down your emergency disk and use it to

HOW UNDELETING AND UNDELETING WORK

Thank goodness DOS is basically lazy. Instead of completely expunging a file, it merely deletes the first character of its name and updates the fat and root directory to mark the file as deleted. Most of the information about the file and its entire contents remain intact. DOS follows the same general procedure when reformatting a disk, updating only the disk's boot sector, FAT, and root directory. An obligatory "scan" is then made of the rest of the disk just to make sure it's there.

Why the laziness? In a word: speed. If DOS went to the trouble of wiping a file off a disk, the procedure would be more time-consuming. I, for one, have no complaints about DOS's methods. File deletion and reformatting are quick, and DOS's sloppiness opens the door for undeleting and unformatting disks when I make an error in judgment.

The UNDELETE command takes the fractured remnants of a file's directory entry and attempts to reconstruct the file's reference points in the FAT. As long as you try to recover a file before information from other files overwrites the original file's contents, recovery proceeds quickly and without a hitch. You are required to supply only the first letter of the filename, which DOS deleted instead of deleting the entire file and its filename.

If you take advantage of MIRROR's deletion-tracking capability, DOS maintains a log, PCTRACKR.DEL, containing details about deleted files. When you use UNDELETE, it checks to see if deletion tracking is active. If it is, UNDELETE fetches the information it needs from PCTRACKR.DEL. Voila, the file is back. In this case, you don't need to provide the first letter of the filename, because the deletion-tracking file supplies it.

Unformatting a disk is a bit more involved than undeleting a file. For one thing, the disk's FAT and root directory no longer exist. The disk's files and subdirectories remain intact, but they are scattered throughout the disk. It's the UNFORMAT command's job to seek out those files and subdirectories and carefully rebuild the FAT and root directory.

UNFORMAT is equal to the task, but rebuilding the FAT and root directory takes a great deal of time and produces a root directory filled with generic directory names.(Since the root directory was destroyed, the UNFORMAT command doesn't know the names of the directories it rescues.) As with UNDELETE, unless you initiate recovery soon after discovering your mistake, your rescue mission may not prove entirely successful.

If you use the MIRROR command on your hard drive, recovering directories and files with UNFORMAT is a breeze. MIRROR saves a file called MIRROR.FIL at the physical end of the disk, where it is unlikely to be overwritten or zapped by the FORMAT command. The UNFORMAT command can quickly locate MIRROR.FIL, using the information it finds there to restore the disk's root directory, FAT, and boot sector. Because the UNFORMAT command doesn't have to do painstaking reconstruction work, unformatting is nearly immediate.

—D.G.

USING MIRROR, UNDELETE, AND UNFORMAT

ACTION: Backing up a hard disk partition
COMMAND SYNTAX: `MIRROR /PARTN`
DESCRIPTION: Used with the /PARTN switch, MIRROR saves partition information about all your hard drives' partition tables in a file named PARTNSAV.FIL. MIRROR prompts you to insert the floppy disk on which you want to save the file.

ACTION: Restoring a hard disk
COMMAND SYNTAX: `UNFORMAT /PARTN`
DESCRIPTION: Using the /PARTN switch with the UNFORMAT partition lets you restore a hard disk's partition table. UNFORMAT prompts you to insert the disk containing the PARTNSAV.FIL file.

ACTION: Tracking deleted files
COMMAND SYNTAX: `MIRROR [drive:[...] [/L] [/Tdrive[-entries][...]]`
DESCRIPTION: MIRROR backs up important information about the boot sector, FAT, and root directory of the drives you name, saving this information in a file named MIRROR.FIL in the root directory. When you specify more than one drive, you must separate them with a space.

The /L switch tells MIRROR to retain only the latest information about each drive, copying any existing MIRROR.FIL file to MIRROR.BAK each time it runs. If you omit this switch, MIRROR makes a backup copy of the existing MIRROR.FIL before storing the new information in MIRROR.FIL.

The /T switch activates MIRROR's deletion-tracking feature. You follow /T with a drive letter (without a colon). If you want to track file deletions on multiple drives, you must use a /T switch for each drive.

The entries option lets you specify how many deleted files MIRROR monitors. Its value ranges from 1 through 999; the default value depends on the size of the disk.

ACTION: Unloading deletion
COMMAND SYNTAX: `MIRROR /U`
DESCRIPTION: The /U switch tells DOS to disable deletion tracking and unload the memory-resident portion of MIRROR from memory. You should unload MIRROR from memory only after unloading any memory-resident programs loaded after it.

ACTION: Undeleting files
COMMAND SYNTAX: `UNDELETE[[drive:][path]filename] [/LIST|/ALL] [/DOS|/DT]`
DESCRIPTION: This command lets you undelete files by specifying the location and names of the files to recover. By default, the command recovers all deleted files in the current directory.

When you use the /LIST switch, UNDELETE provides the names of the files available for recovery.

/ALL recovers deleted files without asking you to confirm that you want to recover each one. If deletion-tracking information is available, UNDELETE uses it; otherwise it uses the DOS directory to recover the files, substituting a pound sign (#) for the filename's missing first character.

/DOS tells UNDELETE to recover only those files that DOS lists as deleted. UNDELETE lets you choose whether to undelete each file.

/DT tells MIRROR to recover the files in the MIRROR.FIL file, asking you to choose whether to delete each file.

ACTION: Unformatting a hard drive
COMMAND SYNTAX: `UNFORMAT drive: /J` or `UNFORMAT drive: [/U] [/L] [/TEST] [/P]`
DESCRIPTION: This command unformats the disk in the drive you specify. The /J switch tells MIRROR to confirm that the information in the MIRROR.FIL file matches what UNFORMAT finds on the drive. The unformatting process is automated and proceeds quickly.

You should use UNFORMAT as soon as possible after accidentally formatting a hard drive. This ensures that you don't write new files over old files that you want to recover.

/U unformats a disk using the root directory and FAT. Use this switch when you know that you don't have any MIRROR files or they are too old to be useful.

Use /TEST to see a recreation of the information that UNFORMAT finds on your hard disk. This switch does not actually unformat your disk.

/P tells UNFORMAT to send its output to your printer.

CONSIDER THE ALTERNATIVES

The MIRROR command included with DOS 5.0 originated with Central Point Software's PC Tools (800-445-4208, $179). That's also where Microsoft snatched (legally, of course) DOS's UNDELETE and UNFORMAT commands. PC Tools still comes with the MIRROR, UNDELETE and UNFORMAT commands, although Symantec's current versions offer more features and performance superior to that provided by their DOS counterparts. But these aren't the only file- and disk-recovery programs on the market.

The very first commercial disk-protection and format-recovery program was RXBAK, one of the programs included with The Mace Utilities. RXBAK copied a disk's boot sector, root directory, and FAT to a special hidden file on your disk. You then could use other utilities within the package to unformat the disk based on the information in the hidden file. Today's version of Fifth Generation Systems' The Mace Utilities (800-873-4384, $149) offers much the same undeleting and unformatting features.

The Norton Utilities from Symantec (800-441-7234, $179) uses a program called IMAGE to create an extra copy of a disk's vital information. Consistently employing the package's SafeFormat program to format disks ensures that you can access this vital information and recover from an accidental reformat. Norton's UnFormat and UnErase utilities use the information created by the IMAGE program to help reconstruct lost data.

—G.G.

boot up. Then, confirm that the drive's partition information really is corrupted by invoking FDISK.COM, one of the programs I recommend putting on your emergency boot disk. Select the Display Partition Information option from FDISK's main menu. If the partition table is indeed damaged, you won't see partition information for the drive in question, or one or more of the entries describing the partition may be garbled. If you see partitions there, and they're DOS partitions or "extended" partitions, which handle partitioning for your system's logical drives, then the partition table isn't the source of your problem. You'll need a DOS guru or a third-party utility, such as The Norton Disk Doctor to help you sort things out (see the sidebar "Consider the Alternatives").

When damage to a partition table is to blame for a hard disk problem, and you have wisely used `MIRROR /PARTN` to save a copy of the partition table, initiate recovery by typing:

```
UNFORMAT /PARTN
```

The /PARTN switch directs the UNFORMAT command to restore your hard drive's partition information using the PARTNSAV.FIL file. When UNFORMAT prompts you for the disk containing that file, insert the proper disk into the drive. After the partition table is restored, reset your PC and the hard drive should boot as it did before.

The corruption of a partition table or boot sector just happens, but the reformatting of a hard drive is self-inflicted. These days, DOS makes you specify the letter of the drive you want to format (something it hasn't always done) and issues three warnings before carrying out your request. But if the big scary-scary happens, and you blunder into reformatting your hard drive, don't despair. UNFORMAT can recover your files, and things will look really rosy if you've recently used the MIRROR command on the hard drive.

By default, UNFORMAT attempts to recover your directory system and its files using information gathered by MIRROR, provided you've been using it. When you type:

```
UNFORMAT C:
```

the program starts by displaying the date and time of the MIRROR.FIL file and its backup file, MIRROR.BAK, and asking if it's OK to reconstruct the hard disk using one of those files. Select the one with the most recent date or, if the most recent one is corrupted, use the backup copy. If the files are too old—or they don't exist, because you haven't gotten around to using MIRROR—UNFORMAT can attempt to rebuild the contents of the disk, using information it finds in the root directory and FAT. Doing this takes longer than rebuilding from MIRROR.FIL or MIRROR.BAK, however, and may not let you reconstruct recently added files or subdirectories.

If you want to verify the accuracy of MIRROR.FIL, issue this command instead:

```
UNFORMAT C: /J
```

The optional /J switch directs UNFORMAT to find out whether the information it finds in the disk's root directory and FAT matches that in MIRROR.FIL.

/J isn't the only switch that UNFORMAT offers for recovering files from a reformatted hard drive (see the table "Using MIRROR, UNDELETE, and UNFORMAT"). But if you use MIRROR to monitor the state of your hard disk, it may well be the only one that you ever need. MIRROR is equally helpful when it comes to undeleting files.

By themselves, UNFORMAT and UNDELETE perform minor miracles on disks, but activating MIRROR makes the procedures for recovering lost information more efficient and more effective. MIRROR is a lifesaver; don't start your PC without it. ■

Free-lance author Dan Gookin resides in Issaquah, Wash. He is the author of numerous books, including DOS for Dummies *(IDG Books).*

Taking Care Of the Environment

Part 3 of this series shows you how to master environment variables and extend DOS's batch language to write complex and useful batch files.

by Hardin Brothers

If you've read the first two articles in this series, you're almost a batch file expert. In part 1, I explained how batch files work and how to use the batch file commands REM, PAUSE, ECHO, and CALL. In the last issue, I looked at SHIFT, GOTO, IF ==, IF ERRORLEVEL, IF EXIST, and FOR...IN...DO, as well as line labels and command-line parameters.

This time, I'll turn to environment variables and begin putting the pieces together to create both simple and advanced batch files. I'll also show you how to use your knowledge of the environment to give your batch files the ability to change to a new directory, carry out a number of tasks, and jump back to the original directory.

Ecology 101

Every program and batch file has an environment—a storage area containing information that each program and batch file receives from the operating system. To see the environment in your computer, type the SET command at the DOS prompt. You'll see a list like the following, although your environment probably will differ and may contain many more entries:

```
COMSPEC=C:\COMMAND.COM
PROMPT=$P$G
PATH=C:\;C:\DOS;C:\WINDOWS
```

The environment is made up of lines of text. Each line has three parts: an environment variable (*COMSPEC*, in the first line above), an equal sign, and text. The equal sign butts up against the text preceding and following it; if you see spaces, you may need to make a correction in your AUTOEXEC.BAT file or another batch file to remove them.

You use the SET command to add a line to the environment, view its contents, or remove the information it contains. For example, to create a variable called TEST, you can type:

```
SET TEST=XYZ
```

To be sure that the line was added to the environment, simply invoke the SET command. To erase the line, type:

```
SET TEST=
```

The root, or first, copy of COMMAND.COM (the program that interprets DOS commands) controls the master environment. Every time you invoke a program or a new copy of COMMAND.COM, a new and separate copy of the environment is created. Each program or invocation of COMMAND.COM can change its copy of the environment and can pass the copy to programs that it starts, but it cannot change (or even read) the master environment in any way that is documented in your DOS manual. This restriction usually is helpful, but sometimes it can be maddening. You can use "undocumented" methods to read and change the master environment from any program, but the methods differ from one version of DOS to another.

By default, the master environment has space for 160 characters (256 characters in DOS 5.0). In DOS 3.1 and later, you can change the default setting by adding a SHELL command to your CONFIG.SYS file. Inherited copies of the environment are trimmed to the length required for the existing strings plus enough extra bytes to make the total length divisible by 16. Therefore, inherited copies of the environment usually don't have enough room for additional strings, unless you pad the root environment with a string you can erase.

Last time, I discussed how to use a percent sign (%) within a batch file—in front of a numeral for a command-line parameter and in front of a letter to define a FOR...IN...DO variable. You also can use percent signs around environment variable names, as in:

```
%COMSPEC%
```

When COMMAND.COM executes a batch file and encounters an environment variable name surrounded by percent signs, it looks up that variable in the environment, substitutes the text following the equal sign for the variable name, and then executes the line. For example, if your environment were identical to the example above and a batch file contained the command `ECHO %PATH%`, COMMAND.COM would look for the PATH= line in the environment and interpret the batch file line as:

```
ECHO C:\;C:\DOS;C:\WINDOWS
```

You must remember, however, that the storage buffer COMMAND.COM uses as it works can hold a maximum of only 127 characters. If a line is too long after the substitution is performed, the excess characters are discarded.

Environment variables are important tools in batch file programming. You can use them, for example, to store subdirectory names, command-line parameters, filenames, and other information needed later in the same batch file or in a different one. Let me show you just how useful they can be.

Setting Examples

Many users create a batch file for each of their important applications. Each batch file sets up the path and anything else the application needs, changes to the correct data directory, and runs the necessary program. When the program ends, the batch file cleans up everything and returns the user to the root directory. For example, a batch file for a hypothetical word processor called WPROC might look something like this:

```
SET OLDPATH=%PATH%
PATH C:\WPROC;C:\DOS
CD C:\WORDDAT
WPROC
SET PATH=%OLDPATH%
SET OLDPATH=
CD C:\
```

The first line saves the current path settings in a new environment variable called OLDPATH. The second line creates a new path tailored for the word processor; it contains the subdirectory for the word processor plus the DOS subdirectory (in case you want to exit to DOS from the word processor and run a DOS command). The third line changes to the correct data subdirectory, and the fourth launches the word processor. The last three lines clean up by restoring the original path, erasing the OLDPATH environment variable, and moving back to the root directory once you leave the word processor.

This could be a useful batch file, but it has flaws that may annoy you. It always puts you into the same subdirectory, and it can't load a file into the word processor automatically. In addition, you may want to load a memory-resident utility, such as a label printer or special spelling checker, when you start the word processor. You also may want to unload this special utility from memory when you end. You can satisfy all these requirements in several ways. Here's the batch file I use:

```
UTILITY /L
SET OLDPATH=%PATH%
PATH C:\WPROC;C:\DOS
IF NOT EXIST %1\*.* GOTO PICK_DIR
CD %1
WPROC %2 %3 %4
GOTO CLEAN_UP
:PICK_DIR
NCD
WPROC %1 %2 %3
:CLEAN_UP
SET PATH=%OLDPATH%
SET OLDPATH= UTILITY /U
CD C:\
```

This batch file begins by loading a utility program, called UTILITY, into memory. (Substitute the command required to run your utility program for this line.) Then it saves the path and creates a new one. The next set of lines tests whether I supplied the name of a

Program Listing 1. INPUT.SCR. This Debug script creates the executable program INPUT.COM, which retrieves keystrokes typed by the user.

```
A
SUB  AH,AH
INT  16
OR   AL,AL
JNZ  10E
MOV  AL,AH
OR   AL,80
JMP  0118
CMP  AL,61
JB   0118
CMP  AL,7A
JA   0118
SUB  AL,20
MOV  AH,4C
INT  21

N INPUT.COM
RCX
1C
W
Q
```

End

subdirectory (as the first parameter, %1) after invoking the program name. If so, the batch file changes to that subdirectory and starts the word processor, using any additional command-line parameters (%2 %3 %4) I might have supplied when invoking the batch file.

If the first command-line parameter isn't the name of a subdirectory, or if no parameters exist, the section labelled PICK_DIR runs a utility program that displays the directory tree and lets you pick a subdirectory that you want to move to. I use the program NCD from The Norton Utilities, but you also can use the Directory Maintenance program from PC Tools or any similar program.

Finally, the section labelled CLEAN_UP ends the batch program by resetting the path, unloading the utility program (substitute the appropriate command for `UTILITY /U`), and moving back to the root directory. You may find other ways to customize this batch file to fit your way of working or the particular applications that you use.

Creating a Menu

As soon as you create batch files such as the one above for two or more application programs, you probably should write a menu program that lets you select among those applications. Here's the outline of the program:

Display a menu of choices.
Get the user's response.
Run the appropriate section of the batch file.
Rerun the menu.

This scenario sounds as if it should work well, until you try to figure out how to get the user's response. DOS doesn't offer a batch command that retrieves keystrokes typed by a user. You do, however, have several alternatives.

The DOS 5.0 manual offers advice on the simplest approach: creating a menu batch file that displays a list of numbered choices and then ends. The user presses the correct number and then presses Enter. This runs a batch file called, for instance, 1.BAT. The batch file ends by running MENU.BAT to redisplay the menu.

This system works but requires many extra batch files. Also, if you have two or more menu programs, you need some way to select the correct 1.BAT, 2.BAT, and 3.BAT files.

A variation on the first method has the user type a command such as `MENU 1` to rerun the same menu program using a command-line parameter (1, in this case) to make the necessary selection. A second possibility is to write the menu program in another language and have it launch the correct batch file or run your application directly. You can, for example, use the SHELL command in QBasic to do this.

Unfortunately, QBasic, or almost any other language, leaves part of itself in memory while running your program, a practice that may leave you with too little room for your program. QBasic, for example, reserves more than 280K for itself when it executes the SHELL command.

A third method relies on the "exit" codes that programs can employ to communicate with batch files, using an IF ERRORLEVEL statement to test the result of the program. Taking advantage of this feature, you can write a program that retrieves a single keystroke from the user and returns the input as an error level. Because a standard enhanced keyboard has only 101 keys, and because a program can return any of 256 exit codes, it should be easy to write a program that translates each keystroke into a unique exit code and returns that code to your batch file.

To keep the input routine small—so that it will load and run as quickly as possible—such programs usually are written in assembly language and appear in magazines as scripts that can be run through DOS's Debug utility.

Program Listing 1 is such a script. You can create it with any text editor, including the DOS 5.0 Edit utility. Make sure, however, that you enter it exactly as shown, including the blank line, and that you save it in ASCII or "DOS text" format. To transform the script into an input program you can execute at the DOS prompt, you must have DEBUG.EXE on your computer system. Save Listing 1 as INPUT.SCR, then, from a DOS prompt, type:

```
DEBUG < INPUT.SCR
```

Program Listing 2. TESTINP.SCR. This Debug script creates the executable program TESTINP.COM, which displays the error level INPUT.COM reports for any key you type.

```
A
SUB  AH,AH
INT  16
OR   AL,AL
JNZ  010E
MOV  AL,AH
OR   AL,80
JMP  0118
CMP  AL,61
JB   0118
CMP  AL,7A
JA   0118
SUB  AL,20
PUSH AX
MOV  SI,014F
SUB  AH,AH
MOV  CX,000A
SUB  DX,DX
DIV  CX
ADD  DL,30
MOV  [SI+02],DL
SUB  DX,DX
DIV  CX
ADD  DL,30
MOV  [SI+01],DL
SUB  DX,DX
DIV  CX
ADD  DL,30
MOV  [SI],DL
MOV  DX,014F
MOV  AH,09
INT  21
POP  AX
CMP  AL,1B
JNZ  0100
MOV  AX,4C00
INT  21
AND  [BX+SI],AH
AND  [DI],CL
OR   AH,[SI]

N TESTINP.COM
RCX
55
W
Q
```

End

(assuming Debug is in a directory mentioned in the PATH statement of your AUTOEXEC.BAT file). You'll see some activity on your screen, then the DOS prompt will appear again. You should have a copy of INPUT.COM in the current directory. If you put the program in any directory mentioned in your PATH statement, you can use INPUT.COM to create a batch file menu in any directory.

How do you know what error level, or exit code, DOS reports for each key? The easiest way is to create a second script and save it as TESTINP.SCR (see Program Listing 2). Then, to create an executable program called TESTINP.COM, type the following:

```
DEBUG < TESTINP.SCR
```

This program displays the error level (as reported by INPUT.COM) for any key you type. Just run TESTINP.COM and press keys, recording the number that appears for each key. When you finish, press Esc. Notice that both programs convert lowercase letters to uppercase. Also, neither program reacts to the F11 or F12 key.

Program Listing 3 provides the bare bones of a batch menu program that displays a menu on screen, waits for you to press a key, and branches to the section of the batch file that handles that keystroke. You can use the structure provided as a template to create your own menu programs.

Program Listing 3. Use this sample batch menu program as a template for creating your own batch file menus.

```
@ECHO OFF
REM
REM  A short demonstration of a batch menu program.
REM  Use this as a template to help build your own
REM  menu program.
REM
:START
CLS
ECHO.
ECHO  The Batch Menu
ECHO.
ECHO  1. Display a directory
ECHO.
ECHO  2. Display the current DOS version
ECHO.
ECHO  3. Check the disk drive for errors
ECHO.
ECHO  4. Exit to DOS
ECHO.
ECHO  Enter your choice (1, 2, 3, or 4):
INPUT
IF ERRORLEVEL 49 IF NOT ERRORLEVEL 50 GOTO PART_1
IF ERRORLEVEL 50 IF NOT ERRORLEVEL 51 GOTO PART_2
IF ERRORLEVEL 51 IF NOT ERRORLEVEL 52 GOTO PART_3
IF ERRORLEVEL 52 IF NOT ERRORLEVEL 53 GOTO PART_4
ECHO.
ECHO  Incorrect input. Please try again.
PAUSE
GOTO START

:PART_1
CLS
DIR
PAUSE
GOTO START

:PART_2
CLS
VER
PAUSE
GOTO START

:PART_3
CLS
CHKDSK
PAUSE
GOTO START

:PART_4
CLS
REM  Nothing else is needed here; the
REM  batch file simply stops.
```

End

Undoing a CD

As you work with batch files, you will begin to invent and pick up little tricks. You might initially wonder, for example, how you can change to a new directory, run a program, and then return to the current directory, no matter what that directory happens to be. A little thinking, plus a tip or two, makes the task simple. To return to the current directory, you can include a line such as the following in your batch file:

```
CD %OLD_DIR%
```

This line works as long as you can figure out a way to store the current directory in an environment variable named OLD_DIR. To create an environment variable, use a SET command such as this:

```
SET OLD_DIR=current directory
```

That seems possible, but for this approach to succeed, you must be able to insert the name of the current directory at the end of the line. There are two common ways to find out the name of the current directory. You can set the system prompt to a value such as PG so that the current directory name ($P) is displayed at every DOS prompt. Or you can type the CD command without supplying a directory name; COMMAND.COM will respond by displaying the name of the current directory.

When COMMAND.COM—or a utility program—displays text, you can route that text to a file with the DOS redirection symbols. This capability helps to solve your problem. Here are the basic steps:

- Using redirection, create a batch file containing a single SET command in which you store the name of the current directory in the environment.
- Run that small batch file, then erase it.
- Change directories and do whatever needs to be done.
- Return to the current directory using the batch command `CD %OLD_DIR%`.

With this outline, you should be able to understand the detailed instructions in the sidebar, "Returning to a Directory," which offers further advice on writing the batch file. If you follow the directions carefully, you'll have a working example of how you can use batch files to save the name of a directory (or anything else) in the environment for later recall.

This is just the beginning. As you work at your computer, think about what features you would like in your batch files, and try to invent ways to meet your goals. The more problems you solve, the more you'll enjoy working with batch files. ■

RETURNING TO A DIRECTORY

To create a batch file that lets you go to a new directory, do some work, and return to the old directory, follow these steps. First, create a one-line data file called ORIGDIR.DAT, from the keyboard. You can do this directly from the DOS prompt by typing:

```
COPY CON OLDDIR.DAT
SET OLD_DIR=^Z
```

Make sure that you press Ctrl-Z (shown as ^Z above) before you press the space bar or Enter key. If you don't, the directory path you add to the batch file won't appear immediately after the equal sign, and the batch file won't work. After pressing Ctrl-Z, press the Enter key to return to the DOS prompt.

Then, working in your batch file, include the following lines to save the current directory in an environment string:

```
COPY OLDDIR.DAT SETDIR.BAT
CD >> SETDIR.BAT
CALL SETDIR
DEL SETDIR.BAT
```

Now you can change directories and run any commands you wish. When you reach the point in your batch file at which you want to return to your original directory, insert these commands:

```
CD %OLD_DIR%
SET OLD_DIR=
```

When you have a working batch file, run it. When the above lines are executed, you'll find yourself back in the original directory.

—H.B.

TIP FILES TO THE MAX

A directory on a double-density floppy disk can hold a maximum of 112 files; a high-density floppy disk holds 224 files per directory. You can't convince DOS to exceed this limit, even if there's plenty of unused space on your disk. You can, however, get around this limitation by creating subdirectories on the floppy disk and storing files in them. DOS lets you put the maximum number of files in each subdirectory.

—Kay Yarborough Nelson

Free-lance writer Hardin Brothers has been working with and writing about computers for 13 years. He is technical editor for DOS Resource Guide.

How to Talk So Your Modem Will Listen

Solving communications problems is a lot easier if you know how to speak to your modem in the Hayes AT command language.

by John Premack

Modems are complex devices, but understanding how they work is easy. You can clear up many communications riddles simply by learning the AT commands, a set of standard instructions—originally developed by Hayes Microcomputer Products—that your communications software uses to control your modem. This nontechnical tour of the most important AT commands will show you what they are, how they work, and how you can use them to diagnose and solve common problems.

To get an idea of what the AT commands do, try this exercise. Turn on your modem and load your telecommunications software. (If you have an internal rather than an external modem, turning on your computer generally turns on your modem.) Set your software so that you are talking directly to the modem. This sometimes is called a direct, or local, connection.

Procomm Plus, the package I use, automatically starts you in terminal mode. However, many others do not, instead giving you a menu or another kind of interface. You might need to read your manual to find out how to get into local mode. (The sidebar "Local Unions" offers assistance with making a direct connection within a few popular communications packages.)

All modem commands start with the prefix *AT*—which stands for *attention*—and end with a command string, which may include letters, numbers, and symbols (see the sidebar "Frequently Used AT Commands"). For this exercise, type the command *ATZ*. The letter *Z* is the reset command; it resets a modem to its default settings. If all goes well, the letters should appear on screen as you type. When you press the Enter key, the modem will process the command and respond with an "OK" message if it understands your instructions or with an "ERROR" message if it doesn't understand them. If you have an external modem with lights, some of them should blink briefly.

If the letters don't appear as you type them, try typing the command *ATE1* first. The E1 command tells the modem to display (echo) the characters on screen as you type them. (You might also try typing uppercase letters. Some modems don't recognize lowercase letters.)

If the screen stays blank or the characters don't match what you've typed, make sure that your software and modem are set for the same data-transfer rate, which is measured in bits per second (bps). If you have an external modem, check that it is plugged into an electrical socket and turned on. Also check to see that your cable is attached firmly to the modem and the computer's serial port.

If that doesn't clear up the problem, make sure your software and modem are set to the same communications (COM) port. Some computers have more than one serial port. You won't get out of the starting blocks if your modem is plugged into COM2 while the software is talking to COM1.

If the difficulty persists and you have an internal modem, consider shutting down your system and taking a look inside. An internal modem doesn't use external cables or ports, but it does have tiny switches or jumpers that change the number of the COM port the modem will claim. While you're inside, check to be sure the card is properly seated in its slot if it appears to be improperly seated.

Initialization

Once you've gotten the "OK" message and know that your modem and software are talking to each other, you're ready to initialize the modem. Initialization modifies the modem's default settings so that it will work in harmony with your software. You initialize a modem by issuing an AT command followed by a string of instructions. Most communications software does this for you automatically.

For example, when I load Procomm, it sends the following initialization string to my modem:

```
AT V1 X4 S0=0 S7=60 M0 ^M
```

If you use Procomm Plus, you can see this string by typing Alt-S and selecting Modem Options followed by Modem Commands. Let's see what these commands do.

Commands beginning with Q, V, and X control the "critical result" code and "call progress" messages the modem sends to the telecommunications software. My modem, set to V1 and X4, reports a successful connection at 2400 bps by sending a "CONNECT 2400" message to the program. Procomm Plus needs this information to know whether a call has gone through and, if the connection proves successful, at what speed the other modem is operating. An unsuccessful try produces a

LOCAL UNIONS

To type AT commands and send them directly to your modem, you must put your communications program in terminal mode. This isn't always easy; full-featured packages use scripts that automate log-on procedures and thus insulate the user from the program's innards. In some cases, direct communication isn't even possible. Here's how *DOS Resource Guide* managed to get into local mode from a few popular communications packages:

CROSSTALK

Type `GO LOCAL` at the command prompt.

HYPERACCESS

From the Define System Settings menu, select Load and type in a system name. The Load option switches to the system's settings without initiating a call. Then press Alt-C to get the Comm screen.

SMARTCOM EXEC

Hayes makes its users jump through a few more hoops:

- From the Offline menu, choose Update Phone Book.
- Choose Modify.
- Choose Phone Entry. Select any entry in your directory.
- Choose Major Settings.
- Change entry for Type of Connection to *direct*.
- Go to the Online menu and choose the phone entry you've modified. The "OK" message will appear.

PROCOMM PLUS AND QMODEM

Both Procomm Plus and Qmodem automatically put you into terminal mode, although you might have to press the Enter key first.

If your package doesn't let you work in local mode, many shareware and public domain programs will. You can download them from information services and bulletin boards. We can recommend Popterm, a memory-resident terminal program from Microsystems Software. It's small—the most recent version is a little more than 12K long—so you won't have to spend a lot of money downloading it. You can get Popterm directly from Microsystems' BBS (508-875-8009). An older but nearly identical version is available in the Comm Programs library of the IBMCOMM forum on CompuServe. Popterm is shareware—if you like it, be ready to cut Microsystems a check for $39.

Even more basic is Jaxtalk, which, at a spartan 459 bytes, bills itself as "The World's Smallest IBM PC Comm Program." Jaxtalk doesn't give you any choice—you have to use the AT command set. This won't do as your everyday comm package, but it's good enough to let you experiment with the AT commands described in this article. Jaxtalk is a free, public domain program available on CompuServe.

Finally, shareware versions of Procomm are available on many BBSes and information systems.

—Eds.

"BUSY," "NO DIALTONE," or "NO CARRIER" message. The meanings of the first two are obvious. The latter message is a little more vague; it means your modem does not detect a modem at the other end of an open phone line.

The command V0 tells the modem to issue its responses as numbers instead of words: 10 instead of "CONNECT 2400," 3 in place of "NO CARRIER," and 6 instead of "NO DIALTONE." Because Procomm Plus expects to see verbal result codes, the V1 command appears in my initialization string.

Inserting the Q1 command in a setup string shuts off codes. It's useful when a modem is connected to a mainframe or minicomputer network and must stifle all responses, passing only data originating from a remote system. If your communications software doesn't appear to recognize what your modem is doing or know the speed at which a connection was made, you usually can fix the problem with the right Q, V, or X command.

Several other commands can be useful in your setup string. For instance, a modem that is causing problems because it answers incoming calls and tries to go on line whenever the phone rings must be told to ignore incoming calls (especially if you have a teenager in the house). You do this with the command S0=0, setting to zero the contents of the register that tells the modem which ring to answer. If you want the modem to answer incoming calls, you can set S0 to a number greater than zero. S0=2, for example, causes the modem to answer after the second ring.

The command S7=60 prevents my modem from abandoning its attempt at going on line before a call has a chance to go through. A modem with a default wait-for-carrier time of only 30 seconds may not stay connected to the telephone line—*remain off hook*, in telecommunications parlance—long enough for some calls to be completed. The S7=60 command in my initialization string tells my modem to wait 60 seconds before giving up.

You can adjust the volume of your modem with the commands L0 through L3. The smaller the number, the lower the volume. M0 turns the speaker off completely.

The last two characters of the initialization string, ^M (pronounced as *Control-M*), represent the carriage return that ends an AT command. Since Procomm Plus doesn't let you use the Ctrl key to enter ^M, you must type it as two characters, the caret (Shift-6) and the letter M.

When you set up your initialization string, you don't need

FREQUENTLY USED AT COMMANDS

COMMAND	MEANING
AT	Command prefix, which gets the modem's attention.
A	Enter answer mode. Tells modem to "pick up" the phone (go off hook) and try to connect to originating modem. (Must be last command in string before carriage return.)
A/	Repeat previous AT command (AT prefix and carriage return not required).
DT	Dial using touch tones.
DP	Dial using rotary pulses.
,	Pause two seconds (value of register S8) when dialing. (Use with D command.)
W	Wait for second dial tone before proceeding. (Use with D command.)
;	Return to command state after dialing, maintaining connection. (Use with D command; must be last command in string before carriage return.)
E1	Display (echo) commands on screen as they are executed.
E2	Turn echoing off.
H0	Hang up telephone (go on hook).
H1	Connect to telephone line (go off hook).
L1	Set speaker volume to low.
L2	Set speaker volume to medium.
L3	Set speaker volume to high.
M0	Turn speaker off.
M1	Leave speaker on until connected and on line.
M2	Keep speaker on.
M3	Leave speaker on until carrier is detected.
O	Return to data mode from command mode when already on line.
Q0	Display result codes.
Q1	Don't display result codes.
V0	Use numbers as result codes.
V1	Use words as result codes.
X0	Provide basic call-progress reports, without giving connect-speed indication.
X1	Provide basic call-progress reports and connect-speed indication.
X2	Same as X1, also provides dial tone detection.
X3	Same as X1, also provides busy signal detection.
X4	Same as X1, also provides dial tone and busy signal detection.
Z	Reset to default modem settings.
S0=n	Allow *n* rings before answering. (Use S0=0 for no answer.)
S7=n	Wait for carrier signal for *n* seconds.
Sr?	Display current value of register r.

the spaces between commands, but they will make your AT string more readable.

Going On Line

Once you've initialized the modem, you're ready to place a call. The command ATDT is the mantra of many a telecommunicator: It tells the modem to dial a number. For example, the following string calls MCI Mail:

```
ATDT 18004566245
```

The DT command tells the modem that you have a touch-tone line. Use DP if you have a rotary line (the *P* stands for *pulse*). You need to use the *T* or *P* only with your first call of a session. The string *ATD* will do for any subsequent calls. If you want to save keystrokes, simply type *A/*. (Do not press the Enter key.) The modem will repeat the previous AT command.

If you make calls on a phone system that requires you to dial special access codes for outside lines or long distance calls, the W (for *wait*) command comes in handy. Entering ATDT 881 W 1-800-555-1212 tells the modem to dial a long-distance access code (881) and wait for a new dial tone before completing the call. An alternative to the *W* is the comma. The D command pauses for two seconds for each comma you include in the command string. Some phone systems require the extra time to process input.

You can add these dialing modifiers to your modem's repertoire in several ways. You can type them manually, as in the previous example; enter them in your software's dialing directory as part of the phone number; or include them in the dialing command. In Procomm, you can edit the dialing command by selecting the menu choice Modem Command Options.

Once you enter a dialing command, the modem goes to work. It goes off hook, dials the number, waits for an answer, negotiates a transmission rate with the remote modem, and sends call-progress messages to your computer. When you're connected, the "CONNECT 2400" message appears, and the modem switches to data mode. You're on line and ready to telecommunicate. The modem now ignores AT commands, passing everything you type directly to the system at the other end of the phone line.

Once you've begun an on-line session, the modem will stay off hook and in data mode until it loses the remote computer's carrier or you give the command ATH or ATH0 (note the zero) to hang up. This creates a modem catch-22.

When you're on line, you're in data mode; your modem no longer recognizes the ATH0 command (or any other AT command). Fortunately, the folks at Hayes foresaw the need to get a modem's attention while it's in data mode. If you type an escape code consisting of three plus signs (+++), the modem will return to command mode without dropping the connection. This is useful if you want to change an operating parameter in the middle of a session. (It's important to pause for several seconds before and after the plus signs. This allows the modem to distinguish between a wake-up call and a transmission.) Of course, you also can use the escape sequence to regain control of a modem so you can hang up your phone line without unplugging the telephone cord or turning off your modem.

Troubleshooting

Sometimes, despite your best efforts, you just can't get your modem to go on line. When you're having trouble, your best friend is the modem's built-in speaker. The tiny lights on the front panel of external modems also are mighty helpful. By listening to the progression of sounds—and watching the lights if you have them—you get a pretty good idea of whether your calls are going through and being answered by a modem and whether the two linked modems are able to negotiate a connection.

While the function of the power indicator (sometimes labeled *MR*, for *modem ready*) on an external modem is obvious, you'd be surprised how much telecommunications trouble is traced to an unplugged modem or a power switch in the off position. Sometimes the plug connecting the power pack to the jack on the back of the modem may be loose.

The remaining lights report the status of the modem and its telephone connection to a remote system. The DTR (data terminal ready) indicator confirms that your modem is communicating with the telecom software running on your computer. The OH light, which stands for *off hook*, tells you the modem has "picked up" the phone. The DCD (data carrier detect) light signals a data connection, while the SD (send data) and RD (receive data) indicators flicker as data is transferred. Some modems have additional lights to indicate that high connection speeds or hardware-controlled error correction protocols are in use.

One vexing problem that plagues even experienced users occurs because the software's automatic dialing feature either can't convince the modem to place a call or doesn't know that the modem is on line. This usually happens when the software isn't sending the right dialing commands, or the modem's call-progress reports (modem results) aren't sent in a form that the software can understand. If you have an external modem, you can use both the speaker and lights to pinpoint what is wrong.

If calls aren't going through, check to see if the modem is going off hook. If it isn't even picking up the phone, confirm that you are getting a DTR signal. If not, try adding one or both of the commands &C1 and &D2 to your initialization string. Required by many modems, these commands control the status of the DCD and DTR signals. If you have DTR but no dialing, verify that the dialing prefix includes ATD and the suffix ^M. If your calls are completed and then disconnected, the software may not be getting the message. In this case, the call-progress reports are the place to look for problems.

Telecom Tips

If you use your modem to answer calls, don't overlook the usefulness of ATH1. This command takes the modem off hook, making the line busy and thus preventing incoming calls when your computer is doing something else. This alternative to turning off the modem, used by bulletin board system operators when their systems are down, tells callers that the line is busy without causing unnecessary concern that calls are going unanswered because the system has crashed.

If you have call-waiting phone service, you can add *70 to the end of your dialing prefix, or insert it before the phone number to prevent an incoming call from disrupting your connection. Callers will get a busy signal and your on-line session will be immune to interruption.

If you spend time on the road, you'll run into some unique and unexpected challenges. A reporter friend who uses his notebook computer's built-in modem couldn't log into his newsroom's computer when he was in a rural area that still used rotary phones. A second entry in his Procomm Plus dialing directory—the placement of a *P* in front of the telephone number—solved his dilemma by instructing the modem to ignore its preprogrammed DT dialing command and pulse dial the call.

You also might have trouble calling from an office or hotel served by an electronic switching system (ESS). While ESS telephones may look like conventional phones, right down to the standard RJ-11 wall jacks, the telephone instruments often are not standard; they do not generate conventional touch tones when you make a call. Modems, which use self-generated touch tones or pulses to dial calls, are rendered incommunicado.

The solution is to hook up your modem and manually dial the call using an ESS telephone connected to your modem's telephone jack. When the remote modem answers, the command *ATX1D* connects your modem and lets it complete a connection. The trick is the X1 command, which, in addition to eliminating certain call-progress messages, tells the modem not to expect a dial tone. The D command without a phone number manages the rest.

Finally, a word of caution. While AT commands should work in the same way with all modems, a modem advertised as AT-compatible does not necessarily have a command set that is identical to the Hayes command set. There are some manufacturers who assign different functions to a few of the more obscure commands.

The assurance of Hayes compatibility should be taken to mean that a modem accepts commands preceded by AT and that the commands for dialing, answering, and terminating calls are the same. Beyond that, prudent users should refer to the command reference guide that comes with each modem.

But don't worry; chances are you'll never need all of these commands to turn your next encounter with a recalcitrant modem into a successful on-line session. ATH0! ■

John Premack is chief cameraman at WCVB-TV Channel 5, an ABC affiliate in Boston.

TIP

A FILE WITHOUT A NAME

You can doctor a filename so that nobody knows its name even though information about it shows up in your directory listings. The trick lies in making strategic use of DOS's RENAME (REN) command and the extended ASCII character 255 (a blank), which you type by holding down the Alt key and using your keyboard's numeric keypad.

For example, if you want to protect a file named DOC1, you can type this command at the DOS prompt:

```
REN DOC1
```

then press the spacebar, Alt-255, and Enter. After you issue this command, DOC1's name is displayed as a blank when you list the contents of the directory in which it resides. (Try it and see.)

If you want to use this trick on several files, remember that each needs a unique name. You can use double blanks to rename the next file, and so on up to eight blanks and a file extension containing three blanks to protect other files from scrutiny. If you have still more files to "hide," try an obscure ASCII code, such as 219, which produces a solid box character. Other symbols you might want to use are ASCII values 225 (the beta character), 234 (the omega character), and 254 (the square bullet character). Unlike an ASCII 255, however, these symbols give snoopers a visible hint to assist them in tracking down the filename.

To reassign a "standard" name to the file, simply use REN to rename the file using normal characters.

Be warned: If you use this trick a lot, make sure you write down the old name and the new name and slip the paper in your wallet.

—**Kay Yarborough Nelson**

From Bottleneck To Breakneck

You can speed access to your hard drive simply by moving files around. All you need is a disk optimizer.

by Patrick Marshall

The erosion is so insidious that you don't even notice it. But every time you save data to your disk drive, you are slowly crippling your drive's performance.

You're not to blame. The fault lies with DOS and how it saves files. Over time, DOS scatters data all across your hard drive—a chunk here and a chunk there. To read a fragmented file, the drive's read/write head must scurry around gathering the pieces together. And the delays can be significant if you're doing disk-intensive work, such as searching through database files or running Windows.

The solution is to use a disk optimizer. A utility of this type does what DOS won't do: relocates files, eliminates the holes left by erased files, gathers the shards of fragmented files, and puts the mess back together. An optimizer can speed file retrieval and disk searches by as much as 30 percent—even more if you use Windows.

I'll take a look at some of the more popular optimizers and their features. But first, let's study more closely the problem the optimizer solves.

What's the Problem?

When you save a file, DOS lays it down consecutively. This is a nice, neat method—until you delete a file. Then you've got an unsightly hole. Even more embarrassing to the tidy-minded operating system is the problem of adding data to a file. If the next available slot is occupied by another file, DOS has a problem. Moving the subsequent file or files to make room for a few bytes of data would be awkward.

To fix the problem, DOS's designers came up with an ingenious, high-tech version of connect-the-dots. DOS simply puts the additional data in the first free spot it finds on the disk, wherever it may be. (More recent versions of DOS store data somewhat differently, but the effect still is the same.) The result is that pieces of the file reside in many places on the disk. DOS tracks the file and its fragments in its file allocation table (FAT), which is also stored on disk.

File fragmentation is not a problem when your drive is new and has room to save data in contiguous blocks. But as you save, modify, and delete files, DOS finds that fewer such blocks are available. Ultimately, your files become intertwined like an electronic Gordian knot.

An optimizer cuts through that knot and restores order. The read/write head does less work, and you spend less time watching your disk drive's activity light blink as you save and retrieve data.

Just how much speed you gain depends on several factors. First, the disk drive you're using makes a big difference. The slower the drive, the more time the read/write head spends gluing splintered files together. So optimizing a slow disk will bring more speed gain than optimizing a fast one.

File length also makes a difference. An optimizer will have less impact on a drive that has many short files. Your drive is partitioned into fixed-length clusters. The

POPULAR DISK OPTIMIZERS

DISK OPTIMIZER 4.06, $49.95

SoftLogic Solutions Inc.
1 Perimeter Road
Manchester, NH 03103
800-272-9900

Low price and extras distinguish SoftLogic Solutions' Disk Optimizer. A set of utilities SoftLogic dubs Data Guardian lets you recover deleted files and protects deleted files from overwriting, so you can recover them if you change your mind. Another utility allows you to recover if you accidentally format your hard drive.

Disk Optimizer also is unusual in that it supplies a set of file management tools; these let you find files and peek inside them, as well as encrypt and password-protect them.

As for the optimizer itself, it is competent, though underpowered and irritatingly slow. The program performs a read-only verification of data sectors but not the more thorough read-write verification. And Disk Optimizer is a bit awkward. It forces you to use a separate DOCONFIG program to choose the type of optimization it performs—whether directories are ordered and whether to reorganize the entire disk or only fragmented files.

FASTTRAX 4.03E, $70

FastTrax International
880 Ensenada Ave.
Berkeley, CA 94707
510-525-3510

FastTrax doesn't have a fancy interface, and it doesn't throw in a lot of extra utilities. But it is the most flexible and powerful optimizer on the market.

To begin with, FastTrax gives you complete control over specifying the exact order in which files and subdirectories are rewritten to disk, thus letting you put often-used files closer to the beginning for faster access. But FastTrax goes a step further than other optimizers in letting you reserve free space between groups of files, making it more likely that new files will be placed closer to frequently used applications.

FastTrax also has a better way of packing files during optimization. In addition to stacking files in consecutive clusters, it tries to arrange files so they don't fall across the disk's cylinders. It is the only program that recognizes a file that falls across a cylinder boundary as a fragmented file.

FastTrax offers a safe mode to protect against data loss if the power goes out during an optimization, and it lets you verify data when you optimize. It cannot, however, perform surface diagnostics of your drive or mark bad clusters.

Finally, if you have expanded memory to spare, FastTrax can handle disks of any size and with any number of files.

THE NORTON UTILITIES 6.01, $179

Symantec Corp.
10201 Torre Ave.
Cupertino, CA 95014-2132
408-253-9600

And then there's The Norton Utilities. Norton's Speed Disk isn't the fastest optimizer, nor is it the most flexible. It certainly ranks at the top for ease of use, however, and only PC Tools offers more features.

Much of the credit for Speed Disk's user friendliness goes to the program's well-designed, mouse-driven interface. But Speed Disk also offers an intelligent array of predefined optimization techniques, ranging from a full optimization, including customized placement of files and directories, to a speedier option whose only goal is defragmenting free space. (This last feature might prove useful to Windows users, because Windows needs large chunks of free space for efficient disk swapping. But if you're smart, you'll create a permanent swap file for Windows instead.)

In all, Speed Disk offers five different levels of optimization. And there's also a handy "walk map" feature that lets you check the contents of clusters that are marked as unmovable or bad.

Of course, The Norton Utilities offers much more than Speed Disk. Just four of the two dozen valuable utilities in the basic package—Unerase, System Information, Text Search, and File Find—make the program a must-buy even for the average user. You also get utilities that let you recover from an accidental hard disk format and an editor for directory, FAT, and partition tables—pretty heady company for a disk optimizer.

PC TOOLS 8.0, $179

Central Point Software
15220 NW Greenbrier Parkway
Beaverton, OR 97006
800-445-4208

The main reason to turn to PC Tools' optimizing program, Compress, is if you want PC Tools' other applications and don't want to spend the extra money for a more thorough optimizing program.

Compress gives you nearly complete control over what files and directories to place where during an optimization, and the program's mouse-driven interface makes options easy to find and set. Like Norton's Speed Disk, Compress offers a "walk map" feature that lets you peek into files, and PC Tools includes disk diagnosis tools for checking the integrity of your drive and repairing faults, though the program doesn't perform verification checks during optimization.

Awkwardly, Compress requires you to reboot after an optimization. Otherwise, you risk losing your data when you run other applications. And PC Tools doesn't offer a safe mode for optimizing, which means that if you lose power during optimizing, you're likely to lose data.

Even more than The Norton Utilities, PC Tools is attractive, if only because of the extras that come with Compress. You name it, PC Tools has it—from its DOS shell/file manager to its backup program, from its disk cache to its selection of desktop production tools.

PC-KWIK POWER DISK 4.10, $79.95

Multisoft Corp.
15100 SW Koll Parkway
Beaverton, OR 97006
800-759-5945

PC-Kwik Power Disk is a curious combination of power and weakness. Its menu-driven interface makes the program simple to use, and it offers some high-end features, such as a utility for walking the disk map and peeking into clusters to see what files are there. Power Disk is one of the fastest optimizers and can handle more than 32,000 files per drive.

The program also includes an unusual method of optimizing, in which all fragmented files are moved to the end of the disk. Over time, this means that files that don't change—COM and EXE files in particular—tend to end up toward the beginning of the drive, while data files that change often end up toward the back. Although other programs let you do this through specifying file order by extensions, Power Disk does it automatically.

But even though Power Disk also lets you manually specify directory and file order, it doesn't offer much help in doing so—you have to manually create an ASCII file listing the order.

VOPT 3.05, $49.95

Golden Bow Systems
P.O. Box 3039
San Diego, CA 92103
800-284-3269

The disk optimizer in Golden Bow's VOPT won't win any prizes for power, but it gets high marks for simplicity.

The VOPT optimizer offers few switches and no bells or whistles. You can't give priority to files or directories, and you can't leave free space. In fact, the only useful option lets you specify which drive to optimize, and you're not even prompted for that. If you want to optimize a drive other than the default drive, you have to indicate it on the DOS command line.

Using the /F switch at the command line tells VOPT to automatically delete lost clusters and map out any unused, unreadable clusters. But the program doesn't verify the data it writes to disk. If you want verification, you have to turn on the DOS VERIFY command before you optimize.

VOPT is unusual in that instead of writing cluster by cluster, the program moves data one file at a time. The major drawback to this technique is that the program requires as much free space on the disk drive as the largest file to be optimized. If your disk is nearly full or if you have very large files, you'll want to look to another program.

VOPT's great virtue is its combination of low price and bundled utilities. One option—VBENCH—produces a graphic display of your computer's performance on 12 benchmark tests, including microprocessor functions and disk-read tests. For checking your drives, VOPT offers utilities for plotting the seek time of your hard disk and for checking the speed and sector spacing of disk drives. There's also a program that checks for bad clusters and one that checks floppy disks for errors. Finally, you can use VTSR, another of the program's options, to display what RAM-resident programs you've got loaded, where their interrupts are, and how much RAM they occupy.

length of a cluster can vary from computer to computer, depending on how the drive is formatted, but a typical cluster is 2,048 bytes, or 2K. If a file fits into one cluster, it can't be fragmented. If you've got mostly short files, even if they are fragmented, they won't have many pieces. Defragmenting them won't save much travel time for the read/write head. The longer the files on your disk, the better the chance an optimizer will speed disk access. Also, fragmented files on a full disk probably will be scattered across greater distances, causing the heads to take more time finding the pieces.

Finally, the kind of work you do on your computer will determine how much of a difference in speed you see. If you're just loading a file in the morning and saving it at day's end, defragmenting the disk may not be important. But if you spend a lot of time moving from file to file, adding new files, killing old ones, and performing disk searches, you're likely to notice that your optimized drive performs with a bit more zip.

Optimizing offers another, potentially valuable, benefit. If you try to recover erased files with an unerase program, recovery is easier if the files occupy consecutive clusters.

Having files scattered across noncontiguous clusters isn't the only potential problem, however. Placing files on different cylinders also can slow down retrieval.

Imagine a phonograph record with the grooves cut as circles instead of as a continuous spiral that winds from the outer edge to the center. If a file resides on one circle, the needle doesn't have to move—it simply grabs data as the disk rotates. If the file is on several circles, the needle must move to the appropriate tracks. It is delays from moving the "needle," or read/write head, that optimizers try to reduce.

Safely Optimizing

Optimizing isn't something you must do every day. The programs reviewed here let you check how fragmented a disk is, so you can decide when it's time to clean house. Don't hesitate to optimize even if you find only a couple of fragmented files; after the first time, the job will take only seconds. I optimize every logical drive at the end of each day, a process that usually takes less than two minutes for my 360MB drive.

All the optimizers I mention in the sidebar, "Popular Disk Optimizers," do the same basic thing, and they all do it pretty safely—though each wisely cautions you to back up your files before your first optimization. That's because disk optimizers put your system memory and disk through a more intensive workout than they usually get, and faults that hadn't shown up previously may become evident during an optimization.

Aside from that, what could go wrong? Not much. You could, of course, lose data if the program writes to defective sectors on the hard drive, which is why most optimizers run a verification check, either automatically or as an option. If you have an older MFM drive, you'll want to use the optimizer program's verification procedures for checking sectors and refreshing formatting. Generally, IDE, SCSI, and ESDI drives do not require such checks.

You could lose data if you lose power while the optimizer is running. Programs with a "safe" mode (which means they rewrite the file allocation table before deleting data from its old location) are your best bet. Even then, if the power goes out while the FAT is being updated, you could lose data.

All the optimizers mentioned here warn you not to reorganize a disk if a multitasking environment or memory-resident programs are active. Such programs sometimes initiate their own internal calls that could interrupt the optimization process and result in lost data. You also should optimize before executing the JOIN, SUBST, or FASTOPEN commands from DOS. But don't use a disk optimizer if you use a disk compression program such as Stacker. Programs of this type usually have their own optimizers.

So how do you decide which package is best? The main factors for most users are how easy the program is to use, how much control over the optimization process it offers, what utilities come with the program, and price. ■

Patrick Marshall is a contributing editor for InfoWorld *magazine.*

TIP FAVORABLE ATTRIBUTES

You can inadvertently delete an application and its supporting files if you use `DEL *.*`, the command for deleting the entire contents of a directory, in a directory containing program files. You won't be happy if you have to reinstall the program.

To protect all files in a directory containing only program files, assign each file a read-only attribute (+R) by typing this command at the DOS prompt:

```
ATTRIB +R *.*
```

To protect an individual file, type:

```
ATTRIB +R [path]filename
```

You can do this within the DOS shell, too. Simply highlight the file or files you want to protect, choose the Change Attributes option from the File menu, and click on the Read Only option.

—**Kay Yarborough Nelson**

The Perpetual Calendar

This short QBasic listing prints accurate calendars for any upcoming year on a single sheet of paper.

by Antonio and Alfonso De Pasquale

With all the fanfare surrounding C, Pascal, and other popular programming languages, it's easy to overlook QBasic, which is included with every copy of DOS 5.0. But a good reason to choose QBasic over the other, more complicated languages is that it generally is better for quickly writing short but useful programs. One such application is CALENDAR.BAS, a QBasic program that lets you print out a full year's calendar on a single page.

Using the instructions in this article, you can easily type the program into your computer and put it to work right away. The program works for 1753 and any year thereafter. (1753 is the first full year that England and its American colonies used the Gregorian calendar, which was instituted by Pope Gregory XIII in 1582 to restore the vernal equinox to March 21.) It also accounts for leap years.

CALENDAR.BAS is compatible with any dot-matrix printer or HP LaserJet-series printer.

Entering and Running the Program

To type in CALENDAR.BAS, first load QBasic. Next, type in the Program Listing. When you're finished, save it to disk by selecting Save from the File menu and naming the listing CALENDAR.BAS.

To run the program, select Start from the Run menu or press Shift-F5. The screen clears and a greeting appears. The program then asks for a year; if you type a year before 1753, the program prints an error message and asks you to try again. Once you enter an acceptable year, the program performs its calculations and reminds you to make sure your printer is turned on and on line.

When you're ready, press Enter and the program will begin printing out the calendar. It displays a message informing you when printing is finished.

How It Works

CALENDAR.BAS is a straightforward program. The Setup section clears the screen, sets up four arrays (or collections of related data), and initializes the arrays to zero. In CALENDAR.BAS:

- Year is a three-dimensional array containing the actual dates for each day of the year.
- Month$ holds the names of the 12 months.
- Month holds the length of each month.
- Day$ holds the first letter of the days of the week (S, M, T, W, T, F, and S).

The GetYear section prints the greeting, prompts the user to enter a year, and makes sure it is acceptable. The CalcYear section uses a formula known as Zeller's Congruence Law to find the day of the week Jan. 1 falls on for the year specified. That date is used later as a starting point for filling out the Year array.

CalcYear also checks for the presence of a leap year. The program creates a leap year calendar when one of the following conditions is met:

- The year is not a century year and is divisible by 4. (By this rule, 1988 is a leap year).
- The year is a century year and it is divisible by 400 (for example, 2000).

If either condition is met, the variable LY is set to 1.

Next, the FillYear section fills up the necessary arrays. Notice the use of LY to change the number of days in February from 28 to 29 for a leap year. The data is contained at the bottom of this section.

BuildCalendar contains the commands for printing the specified year across the top of the page. It also calls subroutines to print a row of asterisks, the month, and the days of the week. BuildCalendar also displays the calendar months, placing six on each side of the page.

Program Listing. CALENDAR.BAS is a QBasic program that prints out a yearly calendar for 1753 or any year thereafter.

```
' PERPETUAL CALENDAR PROGRAM

Setup:
    CLS
    CLEAR
    DIM Year(12, 6, 7), Month$(12), Month(12),
      Day$(7)
    FOR X = 1 TO 12
        FOR Y = 0 TO 6
            FOR Z = 0 TO 7
                Year(X, Y, Z) = 0
            NEXT Z
        NEXT Y
    NEXT X

GetYear:
    CLS
    PRINT TAB(30); "Calendar Creator"
    PRINT
    PRINT TAB(20); "By Antonio and Alfonso
      De Pasquale"
    PRINT
    INPUT "What is the calendar year you want";
      Year$
    YR = VAL(Year$)

    IF YR < 1753 THEN
        PRINT
        PRINT "Year must be greater than 1752.  ";
        INPUT "Press Enter to try again"; A$
        GOTO GetYear
    END IF

    PRINT
    PRINT "Please make sure your printer is turned
      on and is on line"
    PRINT "Also, make sure the paper is set to the
      top of the form"
    PRINT
    INPUT "Press Enter when you are ready to
      continue"; A$
    PRINT
    PRINT "Calculating dates...please wait"
    PRINT

CalcYear:
    C = INT(YR / 100)
    IF RIGHT$(STR$(YR), 2) = "00" THEN C = C - 1
    D = (YR - (100 * C)) - 1
    IF D = -1 THEN D = 99
    K = 1
    M = 11
    X = (INT(2.6 * M - .2) + K + D + INT(D / 4) +
      INT(C / 4) - (2 * C)) / 7
    G = ABS(X - INT(X))
    F = INT(7 * G + .00001) + 1

    IF (YR / 4) = INT(YR / 4) AND RIGHT$(Year$, 2)
      <> "00" THEN
        LY = 1
        GOTO FillYear
    END IF

    IF (YR / 400) = INT(YR / 400) AND RIGHT$(Year$,
      2) = "00" THEN
        LY = 1
        GOTO FillYear
    END IF

    LY = 0

FillYear:
    FOR X = 1 TO 7
        READ Day$(X)
    NEXT X
    FOR X = 1 TO 12
        READ Month$(X)
    NEXT X
    FOR X = 1 TO 12
        READ Month(X)
    NEXT X
    IF LY = 1 THEN Month(2) = 29

    FOR X = 1 TO 12
        R = 1
        FOR G = 1 TO Month(X)
            Year(X, R, F) = G
            F = F + 1
            IF F = 8 THEN F = 1: R = R + 1
        NEXT G
    NEXT X
    DATA S,M,T,W,T,F,S
    DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
    DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
    DATA 31,28,31,30,31,30,31,31,30,31,30,31
```

The section ends by displaying a message that says printing is completed. The END statement at the bottom of BuildCalendar terminates the main program.

The remainder of the listing consists of the subroutines used to print the borders (PrintStars), months (PrintMonth), and days of the week (PrintWeek). FindMonth calculates the number of spaces needed to properly center the months across the page.

If you're new to programming with QBasic, give CALENDAR.BAS a try. It provides you with a handy little program and, we hope, will teach you something about this useful language. ■

Antonio and Alfonso De Pasquale are free-lance writers and programmers based in Boston.

```
BuildCalendar:
    LPRINT
    LPRINT
    LPRINT SPACE$(36);
    FOR X = 1 TO 5
        LPRINT MID$(Year$, X, 1); " ";
    NEXT X
    LPRINT
    LPRINT
    FOR I = 1 TO 12 STEP 2
        GOSUB PrintStars
        GOSUB PrintMonth
        GOSUB PrintWeek
        FOR Week = 1 TO 6
            LPRINT SPACE$(7);
            LPRINT "* ";
            FOR X = 1 TO 7
                SELECT CASE Year(I, Week, X)
                    CASE IS = 0
                        LPRINT SPACE$(4);
                    CASE IS < 10
                        SPV = 1
                        LPRINT SPACE$(SPV); Year(I,
                          Week, X);
                    CASE IS > 9
                        SPV = 0
                        LPRINT SPACE$(SPV); Year(I,
                          Week, X);
                END SELECT
            NEXT X
            LPRINT SPACE$(2); "* ";
            FOR X = 1 TO 7
                SELECT CASE Year(I + 1, Week, X)
                    CASE IS = 0
                        LPRINT SPACE$(4);
                    CASE IS < 10
                        SPV = 1
                        LPRINT SPACE$(SPV); Year(I +
                          1, Week, X);
                    CASE IS > 9
                        SPV = 0
                        LPRINT SPACE$(SPV); Year(I +
                          1, Week, X);
                END SELECT
            NEXT X
            LPRINT SPACE$(2); "*"
        NEXT Week
    NEXT I
    GOSUB PrintStars
    LPRINT CHR$(12)
    PRINT "Calendar has been printed."
    END

PrintStars:
    LPRINT SPACE$(7);
    FOR A = 1 TO 65
        LPRINT "*";
    NEXT A
    LPRINT
    RETURN

PrintMonth:
    FOR B = 1 TO 12 STEP 2
        IF B = I THEN
            GOSUB FindMonth
        END IF
    NEXT B
    RETURN

PrintWeek:
    LPRINT SPACE$(7);
    LPRINT "*"; SPACE$(3);
    FOR D = 1 TO 2
        FOR D1 = 1 TO 7
            LPRINT Day$(D1); SPACE$(3);
        NEXT D1
        LPRINT "*"; SPACE$(3);
    NEXT D
    LPRINT
    RETURN

FindMonth:
    T1 = LEN(Month$(B))
    T2 = LEN(Month$(B + 1))
    T3 = INT((33 - T1) / 2)
    T4 = INT((33 - T2) / 2)
    LPRINT SPACE$(7); "*";
    LPRINT SPACE$(T3); Month$(B);
    RT = 33 - T3 - T1
    LPRINT SPACE$(RT - 2); "*";
    LPRINT SPACE$(T4); Month$(B + 1);
    RT = 33 - T4 - T2
    LPRINT SPACE$(RT - 2); "*";
    LPRINT
    RETURN
```

End

SUBSCRIBE TO DOS RESOURCE GUIDE 800-238-6580

PC Essentials For Everyday Users
DOS RESOURCE GUIDE
Make Work For YOU!
Speed Up Your Hard Drive
Money Saving Batch Techniques
42 More Great Tips Inside!
GUIDE TO WINDOWS SOFTWARE
BEST PROGRAMS FOR:
DOS RESOURCE GUIDE
PRESENTS
THE ESSENTIAL GUIDE TO DOS
2nd Edition
Batch File Tips and Tricks
MANAGING MEMORY STEP-BY-STEP
COMPLETE GUIDE To DOS Files

SAVE UP TO 25% OFF

THE COVER PRICE

BUCK THE TREND! STAY WITH DOS

Get 6 issues of DOS RESOURCE GUIDE for $23.70.* That's just $3.95 an issue!

Save even more on 12 issues: $44.00.* That's 25% off the cover price!

Understanding DOS is the key to effective PC-computing. But if you've tried to learn DOS on your own, wrestled with its manual, purchased a guidebook, or taken a course, you're probably still frustrated and confused.

Relax, you're not alone!

One estimate puts the number of DOS-run computers in the marketplace at 70 million. But how many DOS experts are there?

Darn few, and here's why.

Since late 1980, Microsoft Corp.'s DOS, the essential software ingredient of the IBM PC-compatible, has changed many times. Along the way, the operating system that began as 22 cryptic commands has grown to about 100 equally cryptic commands, drivers, and programs.

DOS RESOURCE GUIDE explores DOS's hidden secrets, untangles the twisted technical prose of manuals, and tells you in clear language how to use DOS.
In each issue of DOS RESOURCE GUIDE, you'll find articles about using DOS commands, about managing your computer's memory, or maintaining order on your hard disk.

Beginners will find step-by-step help at copying files, creating and removing directories, and practical advice on using DOS's wildcards to cut down on keystrokes.

More advanced users will learn about batch files, which can automate tedious activities, such as moving or backing up large groups of files. And you'll learn to use DOS 5's new QBasic programming language.

Each issue of DOS RESOURCE GUIDE is filled with tips on using DOS, undocumented shortcuts, and batch files you can type in and use.

So, tap all of your computing potential by learning DOS, personal computing's most important—if not its most popular— operating system. Order today! Use the coupon below, or call:
800-238-6580

☐ **YES!** Send me 6 Issues of DOS Resource Guide for $23.70.* Save me a buck an issue!

☐ **YES!** Save me EVEN MORE: I'll take 12 issues of DOS Resource Guide for $44.00.* That saves me 25% off the cover price!

NAME ______________________________
ADDRESS ______________________________
CITY ____________________ STATE ________ ZIP ________
(Check, Money Order, or Credit Card—Payment must accompany your order.)
☐ AMEX ☐ VISA ☐ MASTERCARD ☐ DISCOVER
CREDIT CARD # ______________________________
EXPIRATION DATE: ______________________________
SIGNATURE: ______________________________

MAIL TO: DOS Resource Guide, 80 Elm Street, Peterborough, NH 03458

*In Canada and Mexico pay $29.70/(12 issues $55.00); all other foreign surface mail pay $41.70/(12 issues $78.00); foreign air mail pay $53.70/(12 issues $99.00). Checks and money orders must be payable in U.S. funds, drawn on a U.S. bank. Maryland residents add 5% sales tax. Please allow 6—8 weeks for delivery.

HV8AD

Understanding the Keys to QBasic

A handful of strategic QBasic command structures, including DO...LOOP, SELECT CASE, and FOR...NEXT, opens the door to more advanced programming.

by John Wolfskill

Modern Basic, which includes the QBasic program bundled with DOS 5.0, contains scores of commands, functions, and statements. Although this gives newer implementations of the language a great deal of versatility, sometimes the sheer number of expressions you need to master makes writing a program seem an insurmountable chore. Fortunately, a handful of keywords represent the core of the language; learning to use these essential features effectively will boost your programming skills to a new level.

To take you to this higher ground, this month's programming tutorial explains a few of the most important QBasic structures. At the top of the list are two tools, INKEY$ and SELECT CASE, that permit programs to accept and process information typed at the keyboard. Equally indispensable are DO...LOOP; its variant, DO...WHILE...LOOP; and FOR...NEXT. These three QBasic statements enable programs to perform repetitive tasks, such as counting or printing information on your computer screen.

The addition of the IF...THEN...ELSE statement to the mix further expands QBasic's capabilities. In its various forms, this logical tool allows your programs to make decisions based on a set of conditions, such as the size of a number or the length of a text string.

As I introduce these core QBasic constructions, I'll offer concrete programming examples, sometimes referring to the PPRIMER.BAS printer utility published in the first installment of this series (*DRG*#5). It may help you to have this listing handy. If you have a Basic programming manual, you may want to dig that out, too, so that you can experiment with commands and functions as I mention them.

You Rang?

Most computer programs are designed to interact with the people running those programs, which means they need a way to gather input from the keyboard. QBasic's primary tools for accomplishing this task are the INKEY$ function and the SELECT CASE statement. (I explained another important keyword, INPUT, in Part I of this series.)

Editor's Note: *In* DOS Resource Guide *#5 (page 58), we printed PPRIMER.BAS in its entirety. Limited space, however, precludes our publishing it again here. If you don't have that issue, or Parts II and III of this series,* (DRG*#6 and #7) you may order them for $5.95 each. Call 1-800-238-6580.*

To receive the program code, send us a formatted 5¼- or 3½-inch PC disk in a self-addressed, postage-paid disk mailer. We will copy PPRIMER.BAS onto the disk and return it to you as soon as possible. Address your program requests to: PPRIMER.BAS, DOS Resource Guide, *80 Elm St., Peterborough, NH 03458.*

INKEY$ tells QBasic to check your computer's keyboard buffer to see if it contains a keystroke (or rather an ASCII value representing a particular keystroke) that has not been processed. The box at right shows the statement's basic syntax.

INKEY$

Syntax:

`variable = INKEY$`

where variable is any legal QBasic string variable.

Function:

Checks the keyboard to see whether a key has been pressed, returning the ASCII code of the keystroke if it finds one. If it doesn't find a keystroke, INKEY$ returns an empty string to the program.

If the keyboard buffer is empty, INKEY$ returns an empty string, which QBasic has no information to process, so it executes the next line of code. When the buffer contains an ASCII value, which can range from zero to 255, INKEY$ assigns that value to the variable and removes the keystroke from the keyboard buffer.

To do anything useful with the keystroke, you need to interpret it with a statement such as SELECT CASE or IF...THEN. In QBasic, SELECT CASE can execute one of several blocks of program statements, choosing which block to execute by comparing the ASCII value stored in the variable against the value given in a CASE statement.

The box (below, right) shows the basic syntax for SELECT CASE. (Note that the indentations are for readability; they are not necessary.) It's easier to understand the ins and outs of SELECT CASE, however, if you see a real-world example. The following abbreviated version of a portion of PPRIMER.BAS's main processing loop (section K) should make things clearer (ignore DO and LOOP for now):

```
DO
X$ = INKEY$                      'Check buffer for
                                 'keystroke
 X$= UCASE$(X$)                  'Force ASCII value to
                                 'uppercase
  SELECT CASE X$                 'Top of CASE structure

   CASE "S"                      'If S key is pressed...
    MAXPOS = 26                  'Assign maximum cursor
                                 'column position
    ...                          'More work done here

   CASE "A"                      'If A key is pressed...
    GOSUB SELECTFUNCTION         'Select a program
                                 'editing function
    ...                          'More work done here

   CASE "Q"                      'If Q key is pressed...
    GOSUB SAVEFILE               'Save current data
    ...                          'More work done here

   CASE ELSE                     'Ignore all other
                                 'keystrokes
   END SELECT                    'End of case structure
LOOP
```

SELECT CASE and END SELECT form a shell within which you can place any number of individual CASE statements for evaluation. In this instance, the CASE statements describe the particular keystrokes the printer utility evaluates as input.

As QBasic executes the SELECT CASE structure, it begins by reading the value of the assigned variable, X$. Next, it evaluates the first `CASE` *key* statement to determine if the ASCII value of the S key matches the value stored in X$. If QBasic encounters a match, it executes the block of program statements following the CASE statement. If it doesn't find a match, QBasic checks the subsequent CASE statements. The CASE ELSE clause near the end of this sequence lets your program process keystrokes that do not match any of the CASE statements. Although CASE ELSE is optional, it's a good idea to include it. If you omit CASE ELSE, QBasic displays an error message if none of the CASE values matches the test variable.

SELECT CASE is a powerful tool with many options.

SELECT CASE

Syntax:

```
SELECT CASE(variable)
   CASE value
   statement
...

   CASE value
   statement
...
   CASE ELSE
   statement
...

END SELECT
```

Function: Interprets a keystroke value returned by an INKEY$ statement, comparing it against a value named in one or more CASE statements. If it finds a match in a particular CASE statement, QBasic carries out the action indicated in the statement (or statements) following the CASE statement.

Because I needed to evaluate only standard alphabetic keystrokes in PPRIMER.BAS (section K), I used the `CASE "letter"` variation of the syntax, which saved me from having to look up the ASCII values of the keystrokes. If you want to check for a nonalphabetic key, you must use a different syntax. In section O, for example, I used `CASE CHR$(13)` and `CASE CHR$(27)` to check for the Enter and Esc keys, respectively; `CASE LEFTKEY$` and `CASE HOME$` check for the left-arrow and Home keys, respectively.

Section P shows yet another syntactic variation:

```
CASE CHR$(65) TO CHR$(76)
```

Including the optional TO keyword tells QBasic to execute the block of program statements that follow when the value of the keystroke falls within the stated range.

Jumping Through Loops

As versatile as INKEY$ and SELECT CASE seem for situations in which you want to gather and process input from the keyboard, they can do little real work on their own. To use them effectively, you need to team them with a looping structure. You can accomplish this neatly by adding a DO...LOOP statement like that shown in the box below.

Here's a simple example of a DO loop that "polls" the keyboard, placing any keystroke it finds in the keyboard buffer:

```
DO              'Top of polling loop
 X$ = INKEY$    'Check the keyboard buffer
LOOP            'Bottom of polling loop
```

Unfortunately, the loop dooms the statement within its confines to a purgatorial endless loop, because the sequence offers no way to exit the loop. As a result, the code is useless as is.

My earlier—and more complex—example from section K of PPRIMER.BAS demonstrates a more useful application for DO...LOOP. When this routine is executed, QBasic acts on the statements within the loop, looking for a keystroke and checking to see if the keystroke matches one named in a CASE statement. If it finds a match, QBasic is sent to the appropriate subroutine; if QBasic finds neither a match nor a keystroke, it begins executing the statements within the loop again. The EXIT DO statement then causes the program to exit the loop.

DO...LOOP

Syntax:

```
DO
 statement
LOOP
```

Function: Permits a program to repeatedly execute a block of program statements while (or until) a condition becomes logically true.

IF...THEN...ELSE

Syntax:

```
IF variable operator THEN action {ELSE
action}
```

Block Syntax:

```
IF variable operator THEN
   statement
   ...
   ...
ELSE
   statement
   ...
   ...
END IF
```

Function: Provides a means of evaluating a condition and taking further action based on the results of the evaluation.

Meeting Your Conditions

SELECT CASE isn't the only construction that works well within a DO loop. Pairing a loop with the conditional statement IF...THEN...ELSE also provides real programming power.

IF...THEN structures let your program execute a statement or branch to another section of the program when a condition is met. The condition typically involves the evaluation of a variable (or number) and a relational operator, such as a greater than (>), less than (<), or equal (=) sign (see the box above).

Here's a concrete example of IF...THEN at work:

```
IF COUNTER > 100 THEN EXIT DO
```

Embed this statement in a DO...LOOP, and you have the start of something helpful:

```
DO
 PRINT "I will repeat this for a long time!"
 COUNTER = COUNTER + 1  'And I will increase the
value by one
 IF COUNTER > 100 THEN EXIT DO 'When I reach 101,
bail out!
LOOP
```

Each time QBasic executes the loop, it prints the message "I will repeat this for a long time!", increases the value of the variable Counter by one, and checks the value of Counter. During this check, *if* Counter's value is greater than 100, *then*, QBasic exits the loop. (The EXIT DO statement tells QBasic to exit the loop and continue with the program instruction that immediately follows the key word *LOOP*.)

As you begin writing programs, you'll often find yourself teaming loops and IF...THEN statements. In Section Q of PPRIMER.BAT, I used IF...THEN statements nested within a DO loop to send control strings to the printer. In section O, IF...THEN constructions nested within a SELECT CASE statement, which is in turn nested within a DO loop, handle the program's string editing chores.

A more sophisticated use of the command, however, may require you to include the optional ELSE portion of the statement. For example, the following line tells your computer to sound a high-pitched warning tone when the value of the variable Menuitem is greater than 6:

```
IF MENUITEM > 6 THEN SOUND 880,1 ELSE SOUND 440,1
```

It sounds a lower tone when the value of Menuitem is less than 6. Omitting the ELSE portion of the structure definitely changes the line's function.

Of course, conditional constructions aren't always as simple as this one. When your program must execute several program statements as a result of passing a conditional test, (or when the test variable must pass several conditional tests), you should use the block variation of IF...THEN...ELSE shown in the box.

When the variable passes the test imposed by the operator, the block of statements immediately following THEN is executed, and QBasic executes the line immediately following END IF. If the variable fails the test imposed by the operator, the statements after the ELSE keyword are executed, and QBasic executes the line after END IF. The END IF statement serves as a marker to define the limits of the IF...THEN...ELSE block, visually separating the ELSE block.

Another way to accomplish complex tasks involves nesting a block of IF...THEN...ELSE statements—perhaps to perform variable checking. The routine below, for instance, detects when a user presses a mouse button and whether the click was made within a box located at a predetermined screen location:

```
DECLARE FUNCTION GETMOUSECURSOR(ROW, COLUMN, BUTTONS)
...
...

DO
 GETMOUSECURSOR(ROW, COLUMN, BUTTONS)

 IF (BUTTONS AND 1) THEN
  IF COLUMN > 200 AND COLUMN < 225 THEN
   IF ROW > 100 AND ROW < 135 THEN
      GOSUB GETMENU
    END IF
   END IF
 END IF

LOOP
```

Within the DO loop, this routine first calls the function `GETMOUSECURSOR(ROW, COLUMN, BUTTONS)`, which returns the current row and column positions of the mouse cursor. GETMOUSECURSOR also sets the variable Buttons to 1 if the user is clicking the left mouse button.

Next, the outermost IF...THEN statement determines if the user is clicking the left mouse button. If Buttons doesn't equal 1, the routine bypasses the remaining IF...THEN statements and executes the outermost END IF statement. The user isn't demanding the program's attention, so there's no need to continue the evaluation process.

If the value of Buttons is 1, the routine moves inward to the next IF THEN statement, which compares the column position of the mouse cursor with the minimum (200) and maximum (225) column coordinates of the box. If Column is greater than 200 and less than 225, the evaluation continues. A failure causes the program to bypass the remaining IF...THEN statement, because the user apparently is clicking the mouse button at the wrong screen location.

If the column-coordinate test is successful, QBasic moves inward to the remaining IF...THEN statement, which compares the mouse cursor's row position against the minimum (100) and maximum (135) row coordinates prescribed for the box. If the position of the mouse's cursor falls in the correct range, the program executes the GETMENU subroutine. As before, a failure causes the program to bypass the last IF...THEN statement, because the user is clicking at an acceptable column location but at an unacceptable row location.

The nested IF statements are necessary, because they ensure that the program can call the GETMENU subroutine only when *all* values (for Row, Column, and Buttons) pass *all* the conditional tests. If a further—and less complex—example of nested IF statements would help, refer to the beginning of section M in PPRIMER.BAS to see how I used them to test for the Esc key and the user's action and control strings. The comments should help you understand what is going on. The best way to evaluate—or build—nested IF...THEN statements is to start with the outermost statement and work logically toward the core, where the statement the program ultimately must act on resides.

Whiling Away Time

Now that I've introduced the essentials of IF...THEN,

DO...WHILE
Syntax:

```
DO WHILE condition
  statement
LOOP
```

Function: Provides a means of repeating a statement or series of statements as long as a condition is true.

I'd like to backtrack a bit to show what this structure can accomplish when you use it with a variant DO loop construction, DO...WHILE. Pairing the two extends QBasic's powers, providing a means of terminating a loop when the value of a variable within the loop passes a conditional test. To complete the construction, you need to use a numeric value and a relational operator to test the condition of the variable being manipulated by the loop (see box above).

This simple example shows DO...WHILE and relational operators at work:

```
DO WHILE COUNTER < 11
 PRINT "I will repeat until COUNTER reaches 10!"
 COUNTER = COUNTER + 1          'I will count to 10!
LOOP
```

The following variation accomplishes the same end, but this time the conditional check is performed at the bottom of the loop:

```
DO
 PRINT "I will repeat until COUNTER reaches 10!"
 COUNTER = COUNTER + 1          'I will count to 10!
LOOP WHILE COUNTER < 11
```

Placing a conditional check at the bottom of a loop lets you include certain types of program data, such as sorting routines, that must be examined at least once to determine if action must be taken. In this sequence, for instance, using WHILE at the top of the loop doesn't work:

```
LASTITEM = 100

DO WHILE LASTITEM < 100
  FOR SORTDATA = 1 TO LASTITEM - 1
     IF S(SORTDATA) > S(SORTDATA - 1) THEN
        SWAP S(SORTDATA),S(SORTDATA - 1)
     END IF
  NEXT
  LASTITEM = LASTITEM + 1
LOOP
```

Because the initial value of LASTITEM is greater than the test value, the DO loop never executes.

But placing the conditional test at the bottom of the loop, as in this example, ensures the statements within the loop are executed at least once, even if the initial value of the test variable is higher than the value you are testing for:

```
LASTITEM = 1000

DO
   FOR SORTDATA = 1 TO LASTITEM - 1
     IF S(SORTDATA) > S(SORTDATA - 1) THEN
        SWAP S(SORTDATA),S(SORTDATA - 1)
     END IF
   NEXT
 LASTITEM = LASTITEM + 1
LOOP WHILE LASTITEM < 100
```

Next, Please

So far, I've demonstrated how to glean input from the keyboard and process that information with help from SELECT CASE, DO...LOOP, DO...WHILE, and IF...THEN. But no discussion of key QBasic structures is complete without the mention of one the most commonly used data processing statements: FOR...NEXT. Like IF...THEN, this construction is useful for performing a series of instructions until the value of a counter variable exceeds or falls below a stated value (see the box on page 56).

This example from section S of PPRIMER.BAS shows how you might use this format:

```
FOR J = 6 TO 17 'Top of counter loop
 LOCATE J, 5    'Position the cursor
 ...            'Do something more
NEXT            'Bottom of counter loop
```

The initial value for the loop counter is established by the starting counter, 6, as the program enters the loop. The value of the ending counter, 17, determines the number of times the program executes the loop. With each pass through the loop, the number increases by one. Immediately after the number is increased, QBasic compares its value to the ending value, looping through the sequence again if the counter value is less than or equal to the stated ending value. When the value of the counter exceeds 17, the loop terminates, and the program continues with the instruction following the keyword NEXT.

You may place as many program statements between the lines that contain the FOR and NEXT keywords as you need to accomplish the task at hand. The FOR...NEXT loop eventually produces a result you can present to the user as output.

QBasic offers many variant forms of FOR...NEXT loops. This sequence, for instance, is equivalent to the previous one:

```
LOW = 6      'Assign LOW as the starting value
HIGH = 17    'Assign HIGH as the ending value
```

```
FOR NUMBER = LOW TO HIGH    'Assign NUMBER as the
                            'counter variable
 ...                        'Do something with NUMBER
NEXT                        'Loop until value of
                            'NUMBER is
                            'greater than HIGH
```

Only the value of NUMBER changes each time the looping executes; the values assigned to LOW and HIGH remain unchanged.

By adding the optional STEP keyword, then assigning a value to the variable INCREMENT, we can tell the loop to count forward in steps larger than one.

```
LOW = 6
HIGH = 17
INCREMENT = 2

FOR NUMBER = LOW TO HIGH STEP INCREMENT
 PRINT NUMBER;
NEXT
```

By reversing the positions of the LOW and HIGH variables, then adding a minus sign to the STEP keyword, we can make the loop count backwards by the value of the INCREMENT.

```
INCREMENT = 2

FOR NUMBER = HIGH TO LOW STEP -INCREMENT
 PRINT NUMBER;
NEXT
```

No matter which way you use STEP, remember that the program exits the loop only when the final value of the counter is one step larger than the ending counter when you are increasing the counter. The final counter value is one step smaller than the ending counter when you are decreasing.

You also can use a FOR...NEXT loop to increase or decrease the value of a variable within the body of the loop. Here's a simple example:

```
COUNTER = 1     'Assign an initial value

FOR NUMBER = 1 TO 10
  COUNTER = COUNTER + 1
NEXT
```

These lines, from PPRIMER.BAS's INITSCREEN routine (Section Y), provide a more practical example:

```
 SPOS = 1

FOR LNE = 6 TO 17  'Top of counter loop
 LOCATE LNE, 1     'Locate cursor
 PRINT CHR$(186); MID$(NUM$, SPOS, 2); 'Print part
                                       'of string
 PRINT CHR$(179) +
 STRING$(22, 32) + CHR$(186); 'Print the rest
 SPOS = SPOS + 2.  'Increment counter
NEXT
```

The FOR...NEXT loop above prints part of PPRIMER.BAS's input screen to the display. It's very important to establish the starting string position (the value of SPOS) prior to entering the FOR...NEXT loop. If you omit this line, the program assumes a starting value of zero the first time the loop executes. The resulting string position would be invalid, and QBasic would eventually halt the program, printing "Illegal function call" messages on the screen.

Not all FOR...NEXT loops are as simple as this one. Often, you'll need to nest a loop within another loop to solve a problem. Nesting is among the most useful of QBasic's data processing tools, but it also is one of the most difficult concepts for beginners to grasp.

For instance, a graphics program that displays a complex set of visual data may use nested loops to print the same information at various points on the screen. The outermost loops primarily manipulate the screen display's coordinates, while the innermost loop displays the data on screen. This structure lets the program perform a great deal of work using a relatively compact processing routine. The following nested FOR...NEXT routine demonstrates the technique:

```
FOR ELEMENT = 1 TO 4                 'Print string
                                     'elements 1-4
 FOR ROW = 12 TO 15                  'Print the strings
                                     'on rows 12-15
   FOR COLUMN = 10 TO 70 STEP 10     'Make several
                                     'duplicate columns
     LOCATE ROW, COLUMN              'Locate cursor at
                                     'column 12
     PRINT A$(ELEMENT);              'Print the string
   NEXT
 NEXT
NEXT
```

FOR...NEXT

Syntax:

```
FOR variable = start value TO end value
  program statements
NEXT variable
```

Function: Provides a means of executing a program loop a specific number of times.

Note that the innermost loop controls the action, ending by printing the string. The outer loops play a supporting role by advancing to the next element in the string array.

To figure out a seemingly complex set of nested FOR...NEXT loops, you need to take the opposite tack from the one I recommended with nested IF...THEN statements. Always start at the innermost loop and work outward. Similarly, when you design a routine containing nested loops, first decide what the routine must accomplish, building the innermost loop first and then adding the appropriate outer support loops to allow the central loop to manipulate data efficiently.

Keep It Simple

This overview of key QBasic programming structures just scratches the surface of what you can accomplish with the likes of INKEY$, SELECT CASE, DO loops, IF...THEN, and FOR...NEXT. The more you use these expressions, the more confortable you'll feel with QBasic—especially if you keep things simple at first.

I've said it before: One of the best ways to learn to program is to study sections of other programmers' code. Don't be afraid to play around at the QBasic prompt, typing in bits and pieces from your Basic manual, books, and magazines. Familiarizing yourself with the tools of the trade will impart both a knowledge of the language's fundamental structures and confidence in your own programming ability. ■

John Wolfskill is a free-lance writer and president of Powertrain Development, a New Hampshire-based software development company.

TIP XCOPY ASKS FOR A DISK

You could wish for a faster or less cumbersome method of backing up files than BACKUP, but DOS 5.0 doesn't grant you one. XCOPY, for example, is faster, but it isn't always safe.

The problem with using XCOPY to back up or transfer a directory full of files from your hard disk to floppies is that it doesn't halt its activity long enough to warn you that your data disk is about to fill up. If only the command would stop, request another disk, and then continue its chores.

Well, with some fancy finger work, it can. Because XCOPY includes options to alter a file's archive bit (which tells DOS if a file has been altered since it was backed up last), you can use the command in a short batch file to back up hard disk files to floppies.

Knowing this, I created the following program to assist with backup chores:

```
@ECHO OFF
ATTRIB +A *.*
:LOOP
XCOPY *.* A: /M
IF NOT ERRORLEVEL 2 GOTO END
ECHO Insert next disk
PAUSE
GOTO LOOP
:END
```

This program sets the archive bit for all the files in the current directory. It then enters a loop that executes XCOPY with the /M switch, which accomplishes two things. It tells XCOPY to act on only those files that have their archive bits on (that is, those that have been changed) and, once the file is copied successfully, it turns off the archive bit.

The IF NOT ERRORLEVEL line causes the batch file to stop after all files are copied. The program exits the loop when XCOPY returns an error level of either 1 or zero. The former indicates that no files were found to copy; the latter indicates that all the files were copied without error. If the program doesn't have room for a file, it prompts you to insert another floppy disk.

Unfortunately, however, this approach presents problems. First, it may not work with all versions of DOS because of enhancements in XCOPY's reporting of error levels. The batch file also fails if the file you are copying is larger than the capacity of the target floppy disk. (The program drops into an endless loop, and you must press Ctrl-Break to get out of it.)

Also, this method wastes disk space. For example, if you're copying three files to a 360K floppy disk and two of the files are 150K and the other is 320K, you'll need two target disks. If the first file is 150K, the second 320K, and the third 150K, you will need three target disks. This is because XCOPY copies files in the same order it encounters them on your disk.

—Todd Martin

Expanding Windows: Adding a New Program

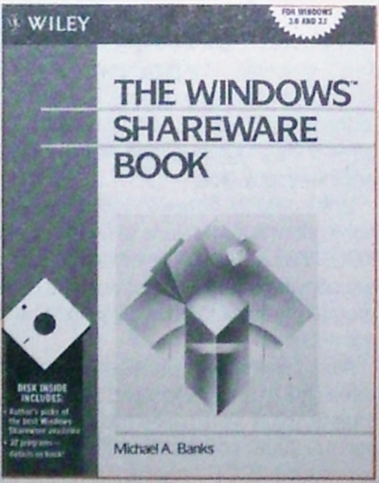
Adding a program to Windows is smoother, if you understand two of its most important tools: Program Manager and Program Manager's File menu.

by Michael A. Banks

Windows Program Manager and the Program Manager's File menu provide the tools by which you expand Windows. This excerpt will explain what Program Manager is, and how to use the options on the File menu to expand Windows. Along with this, other pertinent information will be provided.

To use the information, you already should have a working knowledge of Windows commands and menus. Just in case you're still a little new to Windows, here are some basic things to remember:

- The menu structure is the same in any Windows program, beginning with the File menu at the extreme left and moving to more specialized menus to the right.
- Menus are accessed by moving the mouse pointer to the menu name and clicking, or by pressing Alt and the first letter of the menu name (Alt-F for the File menu).
- To select an item on a Windows menu, point and click with the mouse, or press the underlined letter in the item's name.
- Selecting an item from a list (such as a file on which a command is to act) is done by double-clicking the left mouse button with the mouse pointer on the item name.
- Alternately, you can enter an item name in a text field, when one is provided, as in dialog boxes. To implement a command or action per the items in a dialog box, press Enter or click on the OK button.
- To further refine your selection, click on appropriate check boxes or radio buttons to select or deselect a box's specification. An *X* in the box means the specification is active; an empty check box means the specification is not active. Radio buttons operate in a similar manner: If the circle has a dot in it, the specification



Editor's Note: *The following is excerpted from* The Windows Shareware Book, *by Michael A. Banks (copyright 1992). Reprinted by permission of the publisher, John Wiley & Sons Inc. All rights reserved.*

is then made active.

- To cancel a pending action from a dialog box, undisplay a menu, or back out of most operations, press Esc. You can click on a Cancel button if one is provided.
- You can always get specialized or general help with a program or Windows by pressing F1.

I. USING PROGRAM MANAGER

Windows Program Manager handles a number of vital functions having to do with starting, installing, modifying, and deleting programs on the Windows desktop. Program Manager is also the point at which you exit or quit Windows and return to DOS. Program Manager is, in effect, "command central" for Windows—although some of the utilities you'll learn about in this article can change that.

What Program Manager Does

Program Manager is the first thing you see when you start Windows. It displays all the program groups and programs you have installed, as illustrated in Fig. 1.

The program groups are represented by groups of icons within boxes, and each box carries the name of the group: Accessories, Utilities, and so forth. Icons are symbols (normally with labels) that represent programs. We'll take a closer look at icons in a few paragraphs.

Note that the File menu in Fig. 1 is open. This menu offers several vital commands and options that are especially important because with them, you can install, change, start, delete, and otherwise manipulate shareware and public domain (PD) programs. (You can do the same with existing Windows programs and commercial add-ons.)

Also note that throughout Windows, you'll see some menu items followed by ellipses; for instance, New... . This indicates the item is not a command that is acted on immediately; rather, selecting the item results in Windows asking you for more information, such as a filename, via a dialog box. For example, after you select New... on the File menu, a dialog box appears, along with a prompt to enter information Program Manager needs before it can do anything more.

Other selections not followed by ellipses are acted on immediately, although you may be asked to confirm or verify the command before it is implemented. Some of these immediate commands can be issued via the keyboard. When this is the case, the command is followed by its keyboard equivalent, as with the Del (for the Delete key) following the Delete item on the File menu.

Many Windows commands are also accessible with function keys, and Ctrl-, Alt-, and Shift-function key combinations. The two you'll probably use most frequently are F1 for help and Alt-F4, which exits a program.

Starting a Program from Program Manager

There are several ways to run a program with Program Manager. The simplest (when the desktop is displayed, as in Fig. 1) is to move the mouse pointer to the icon that represents the program in question, and then click the left mouse button twice.

Alternately, you can click on a program's icon once to highlight it, then press Enter. This provides a convenient way to start a frequently used Windows program. If you exit Windows with a particular program within a program group highlighted (and save the configuration by putting an *X* in the Save Changes check box), pressing Enter the next time you start Windows will run the highlighted program.

Similar to highlighting and pressing Enter (though it adds an extra, needless step to the process) is highlighting a program icon and selecting Open on the File menu. (Note that the keyboard counterpart of Open on the menu is the Enter key.)

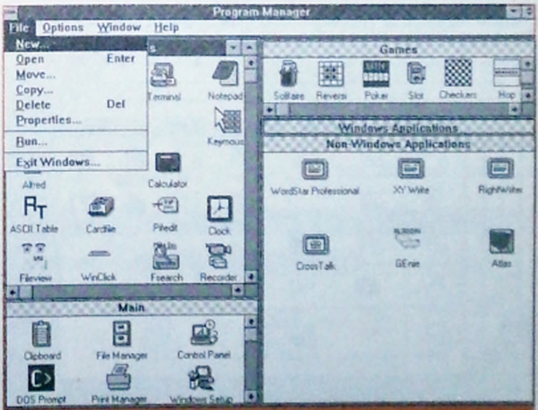


Fig. 1. Windows Program Manager is the main control center for installing new programs.

Yet another way to start a program is to select the Run... option on the File menu. This is normally used to start DOS programs that aren't in a program group (that is, programs that don't have a program information file [PIF]). However, using the File menu's Run... option to start a Windows program involves useless extra steps; the previous two methods of running a program are more efficient.

Program Groups, And Icons

Program groups and icons are integral elements in expanding Windows. This section defines and discusses the role of each.

What's a Program Group?

You'll have noticed that Windows programs (those that come with Windows, Windows versions of commercial programs, shareware and public domain programs, and DOS programs) are organized into groups. Each group is clearly labeled, and the active group's name is highlighted. File menu options act on the highlighted, active program group—which, as you will learn, is very important in terms of expanding Windows.

Program groups can be displayed in several ways. Fig. 1 shows one way the groups can be displayed. (Note that the contents of the Windows Applications group, which contains commercial programs, aren't displayed. However, the window containing these programs can be displayed by clicking on the group title.) This particular arrangement is called "tiled." Another arrangement, called "cascade," is shown in Fig. 2.

Program group display formats are accessible via the Window menu.

It also is possible to expand a program group, as you would any other window, so that it fills the screen. This is useful when you have more programs in a group than can be displayed in a tiled or cascade arrangement.

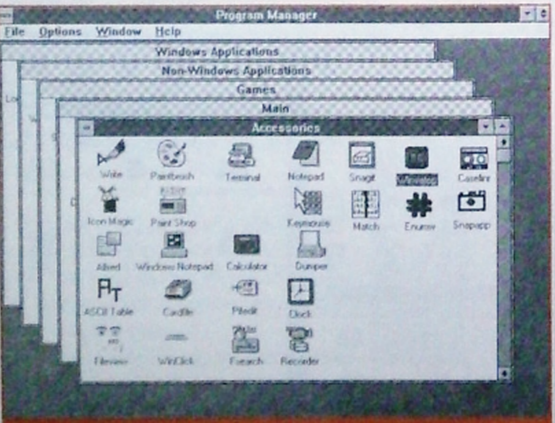


Fig. 2. Windows program groups shown in cascade arrangement.

What's an Icon?

An icon is a graphic symbol or drawing that represents a program on the Windows desktop. Icons make finding an item on the desktop easy. A typical icon is a logo or consists of art that relates to the program in question. In addition, an icon carries a label, identifying the program it represents.

Icons are stored on disk in either of two ways:

- In a file with the filename extension ICO (ALFRED.ICO, for example). When this is the case, you can (and sometimes must) specify that the icon file in question be used with a specific program. Often, the program already is set up to find and use its icon file.
- In a program (EXE) file. When this is the case, the program uses the embedded icon file as the default. Sometimes, two or more icons are included within a program file, and you can select the desired icon during the installation process.

You'll want to have an icon for any program you'll use frequently. In some instances you'll have to create an icon for a program, unless you want to use one of the icons supplied with Windows. However, most shareware and public domain programs come with icons, and there are enormous libraries of icons available from public domain and shareware sources. (If an existing icon doesn't quite suit your tastes, you can edit it, using one of several shareware icon editors available.)

II. ADDING A PROGRAM

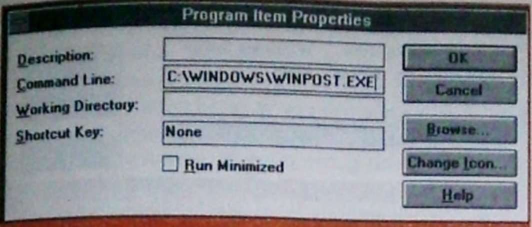
As noted, program groups and icons are the major elements to consider when adding a shareware or public domain program to Windows. The procedure for adding a program is fairly simple. Many Windows programs come with their own installation programs—usually named INSTALL.EXE. When one is available, all you have to do is place the program's files on a floppy disk or temporarily in their own directory, click on Run... on Program Manager's File menu, and then enter the name and location of the installation program. The installation program takes over from there.

It is important to note that you must always click on the Save Changes check box in the Exit Windows dialog box when you exit Windows after making any kind of program addition or change.

If you don't do this, some changes you made won't exist the next time you run Windows. Any programs you may have added will be retained, but their icons will not be where you placed them in the group.

Preparing a Program for Installation

The first step is to make sure a program and its associated files are in the Windows directory on your hard disk. Depending on where you obtained the program, you may have to unpack it from its ZIP, ARC, or ARJ format. The files will include an EXE file (the program



Description:
Command Line: C:\WINDOWS\WINPOST.EXE
Working Directory:
Shortcut Key: None
☐ Run Minimized

Fig. 3. The Program Item Properties dialog box lets you match a program's icon with its associated files.

itself), and may include an icon file (ICO), a DLL file, and perhaps other data files that the program requires. For the sake of this article, we'll be using WinPost, a notepad program included with *The Windows Shareware Book*. The files in the WinPost ZIP file (WPST32.ZIP) are:

- WINPOST.EXE, the program file
- WINPOST.HLP, a help file, accessible when WinPost is active
- WINPOST.WRI, the WinPost user manual in Write format
- README.TXT, new information not covered in the manual
- VENDOR.DOC, information for disk distributors
- PACKING.LST, a text file describing these files

Of these, you need only WINPOST.EXE and WINPOST.HLP to run the program. If you want to be able to read and/or print out the manual, you'll need to have WINPOST.WRI handy.

Placing the Program and Starting Installation

After getting the required files into the Windows directory, highlight the group in which the program's icon will be displayed. (This is necessary because the program you are installing is placed in the currently highlighted group.) Do this by clicking once anywhere within the group's window. When this is done, select New... on the File menu. The New Program Object dialog box is displayed. Select Program Item and press the OK button (or Enter) to display the Program Item Properties dialog box shown in Fig. 3.

Specifying Program Properties

Fill in the Description field with the label that you intend to display beneath the icon on the desktop (in this case, WinPost). In the Command Line field, type the directory path and the name of the program. Because the WINPOST.EXE file is in the Windows directory, no directory path specification is required.

As an alternative to entering the name of the program in the Command Line field, you can press the Browse button and select from a list of available programs in the current directory.

After you've completed these steps, press Enter or select the OK button. The WinPost icon is displayed on the desktop, in the Accessories group, ready to be started like any other Windows program.

Selecting a Different Icon

During the program installation process, you can select an alternative icon (if one is available) by pressing the Change Icon... button. (This also is a quick way to preview the icon the program will use.) The dialog box shown in Fig. 4 will be displayed.

Click on the scroll bar to see alternative icons. When you see an icon you want to use, highlight it with the mouse and click on OK to accept it.

Moving an Icon within a Group

Once the program is installed, you'll probably want to reposition the icon within the program group. Moving it is no problem; simply click on the icon, hold down the mouse button, move the icon to the desired location, and then release the mouse button.

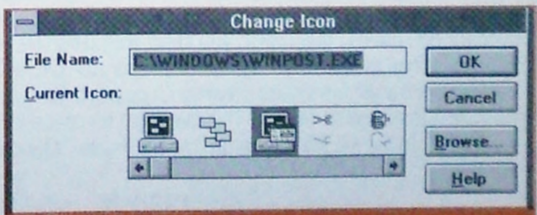
Using Windows Setup to Add Programs

You can use the Windows Setup utility to add a number of Windows programs at once. The utility automatically searches your system and finds all the programs that can be added as Windows programs. Note that the Windows Setup utility is not perfect; it will miss some programs. However, it will find most of the installable programs. (Note, too, that this utility automatically installs most programs in appropriate groups the first time you install Windows and run Setup.)

To start the Windows Setup utility, click on it in the Main program group. Select Setup Applications... on the Options menu. You'll be asked whether the utility is to search the current path, a specified hard drive, or all drives. Once you make this selection, the Windows Setup utility searches the specified drive(s) and prepare a list of programs that you can install. The process from there is fairly self-explanatory; you can have the utility install selected programs, or all the programs on the list, automatically.

III. MAKING CHANGES IN INSTALLED PROGRAMS

There are a number of ways you can change installed



File Name: C:\WINDOWS\WINPOST.EXE

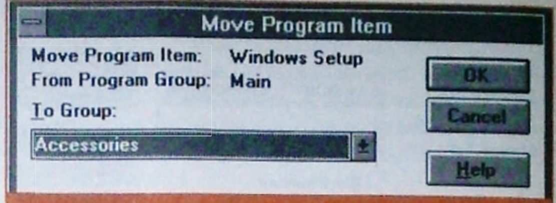
Fig. 4. The Change Icon dialog box lets you choose an alternative icon for the specified program..

programs. As already described, you can move a program's icon to a more practical or aesthetically pleasing location within its group. (You already know about using File menu options to install or start a program.) Other selections on the File menu let you move or copy a program from one group to another, delete a program, or modify it. You also can use the PIF editor to change the way a program runs.

Moving a Program from One Group to Another

If, after you've installed in a group, you decide it really belongs in another group, moving it is easier than you might think. With the mouse, just place the pointer on the program icon you want to move, press and hold the left mouse button, drag the icon to its new location in a different group window, then release the mouse button. You can also move a program using the menus. Simply highlight the desired program's icon by clicking on it once, then select Move... from the File menu. The Move Program Item dialog box shown in Fig. 5 will be displayed.

Select the group to which you wish to move the program, then either double-click on it or press the OK button. The item will be moved to the specified group.



Move Program Item: Windows Setup
From Program Group: Main
To Group: Accessories

Fig. 5. The Move Program Item dialog box lets you move a program from one group to another.

(You probably will have to move the icon to the location you desire within the group.)

Copying a Program to Another Group

You may wish to start a program from more than one program group—perhaps a frequently used application such as Write—without having to display a particular program group or go to the File menu to run it. This often is the case for those who use Windows with one program group displayed at a time.

To be able to start a program from more than one program group, copy it to the desired groups. With the mouse, copying a program item is just like moving it, except that you hold down the control (Ctrl) key while dragging the icon from one group window to another. The icon will remain in the original location and a copy will appear at the new location. To do this with menus, highlight the program's icon, then select Copy . . . on the File menu. In a process similar to moving a program item, you'll be asked to specify to which group you wish to copy the program.

Deleting a Program

To delete a program item, highlight its icon and select Delete on the File menu. Or highlight the icon and press the Del key. You'll be asked to confirm the deletion.

Modifying a Program

If, after installing a program you wish to change one or more of its properties, you can do so by highlighting its icon and selecting Properties... on the File menu. The Program Item Properties dialog box discussed earlier and shown in Fig. 3 is displayed, and you can change the icon or other properties, as desired.

TIP COLOR-COORDINATED DRIVES

If you have a color monitor and more than one hard drive—either physical drives, RAM, or logical drives (partitions, or those created with the SUBST command)—why not create a batch file that sets a different foreground and background color for each drive? When you execute the batch file, it will change you to the drive you specify and alter the screen colors automatically.

Here's an example, called D.BAT, that switches you to drive D and changes the screen colors to a red background with black text:

```
@ECHO OFF
CLS
D:
PROMPT $E[41;30m$P$G
```

(Before you begin, make sure CONFIG.SYS includes a DEVICE directive that loads ANSI.SYS.)

To execute it, just type `D` at the prompt instead of the usual `D:` you type to change to drive D (assuming you keep your batch files in a directory that's specified in the PATH statement in your AUTOEXEC.BAT).

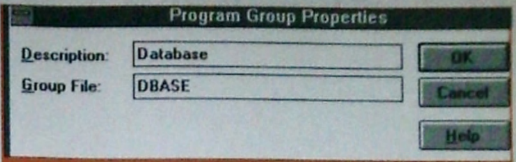
If you want a different color on drive C, create a file called C.BAT and execute it to switch to drive C. This file might contain a PROMPT statement such as the following to return the display to DOS's default colors:

```
PROMPT $E[0m$P$G
```

—Kay Yarborough Nelson

IV. CREATING AND USING PROGRAM GROUPS

You may, on occasion, find it necessary to create a program group, either because there is no room to display all the programs you have in a group in a pleasing manner, or (more likely) because you are adding programs that don't fit in an existing group. When this is the case, you can create a new program group using the New... option on the File menu. The procedure here is similar to adding a new program to a group. After you select New..., the New Program Object dialog box is displayed. Rather than selecting Program Item, as described earlier, select Program Group. The Program Group Properties dialog box (Fig. 6) will be displayed.



Description: Database
Group File: DBASE

Fig. 6. The Program Group Properties dialog box asks you to enter the name for a new program group.

Enter the name for the program group in the first field. Windows will fill in the name of the program group file in the second field. (This is a file that stores information about the group, for use by Windows.) You can create your own name for the group, but omit the filename extension, because the filename extension GRP is added automatically.

After you specify the name, the group appears on your screen. Its size and position can be changed like that of any other group, and it can be displayed overlapping other Windows, if necessary, by clicking on any visible part of it or by selecting it on the Window menu. (It will be added to the Window menu for you.) You can arrange the group to be fully displayed, or perhaps hidden, as I have done with the Windows Applications program group in Fig. 1. Once the program group exists, you can install, move, copy, and delete program items for it as described earlier.

Those are the basics of expanding Windows. You may find it useful to practice adding, moving, modifying, and deleting a small shareware or public domain program, per the instructions here. (Be sure to keep a copy of the program's original files on hand!) And don't forget to click on the Save Changes check box when you exit Windows after adding a program or making any other changes you want to keep. ■

Michael A. Banks has written three dozen nonfiction books and science fiction novels. He is also co-author, with Jerry Pournelle, of Pournelle's PC Communications Bible *(Microsoft Press) and author of* Portable Communications *(Brady Books). To order a copy of Wiley's* The Windows Shareware Book, *call 800-225-5945.*

TIP KEEPING YOUR SCREEN CLEAN

To reduce on-screen clutter, it's sometimes desirable to prevent the results of batch file commands from appearing. For instance, a batch file might copy a few files, but for the sake of appearance you may not want the filenames and DOS's "*x* file(s) copied" message to be displayed. One way to eliminate this uses the redirection operator (>) to send the output of the COPY command to the NUL device, a dummy device that produces no screen output. Here's a typical command looks like this:

```
COPY TEMP1.TXT > NUL
```

When DOS executes this line, it performs the copy operation as usual, but the screen output of the COPY command isn't displayed. You can use this redirection trick with other DOS commands, as well.

If a batch file contains several consecutive commands whose output you want to suppress, DOS offers a more efficient method of suppressing output: teaming up NUL with CTTY, a little-used command that changes the device used to control your system. By default, the CON (console) device—your display and keyboard—handles system control.

To suppress all screen output from a batch file, include the following line at the point where you want screen output to stop:

```
CTTY NUL
```

From that line on, all output from commands in the batch file goes to the NUL device and will not appear on screen. Inserting the following line later in the batch file restores normal operation:

```
CTTY CON
```

You must use care with this technique. In addition to suppressing screen output, CTTY NUL disables keyboard input. If something goes wrong, and the batch file stops before the `CTTY CON` statement executes, your PC won't respond to keyboard commands until you restart the system by pressing Ctrl-Alt-Del.

—Jeff DeTray

Pen Computing Affordable? Write On!

If proponents of pen-based computers are right, the keyboard will soon be passé, and we'll all be scribbling away with electronic styli on notebook-size computer screens as intelligent software translates our hen-scratchings into binary code.

> **Forms are the number one use, especially in environments in which it's difficult to carry a keyboard.**

But to be honest, I've never understood the attraction of pen computers for everyday jobs such as spreadsheets and (especially) word processing. Most people know how to type, at least a little; and even the slowest hunt-and-peck typing is faster than printing or writing. I'd no sooner write this column out longhand—whether on a pen computer or paper—than cook meals on my wood stove. In fact, for many applications, pen computing seems a step backward. (Are Fred Flinstone's stone tablet, chisel, and mallet next?)

Pen computing also suffers from unmet expectations. One of its promises was that sophisticated software would translate any writing into computer-readable form. It just doesn't work that way. You can find software applications to do the job, but they work only if you carefully print characters and "train" the system to recognize your style of writing. And they still makes mistakes. Software that can decipher your regular handwriting might be as far away as computers that understand your speech. In the meantime, most working pen-based software stores what you input as "electronic ink," essentially a graphics picture of handwriting or printing.

There are, however, a number of applications where pen-based computing makes a lot of sense, and where it's already being used. Forms are the number one use, especially in environments in which it's difficult if not impossible to carry a keyboard. Your UPS delivery person has a pen-based computer on which you sign your signature electronically when a package is delivered. Meter readers, doctors and nurses, and police officers also are starting to use pen-based systems.

Pen computers are attractive for certain desktop applications, too, especially drawing, paint, and computer-aided design (CAD) programs, where the pen provides finer control than mice or trackballs. And the abilities of pen computers also are attractive for educational applications, especially with younger children who haven't learned to use a keyboard. Every kid can draw.

Finding a Starting Place

Until recently, the high cost of hardware has depressed pen computer sales and made it difficult for the average user to experiment with the technology. The leading pen-based systems from companies such as NCR, Samsung, and Grid generally sell for $4,000 to

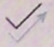
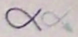
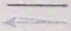
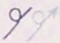
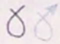
BY STAN MIASTKOWSKI

$6,000, though the cost appears to be coming down slightly. At such prices, it might make more sense to skip the stand-alone pen units and upgrade an existing computer by adding a peripheral device to handle pen input.

A number of alternatives to full-fledged pen computers currently exist, selling generally in the $200 to $600 range. That's still not exactly cheap, but it's a reasonable price for adding an interesting and unique function to your computer. There remains, however, the thorny problem of choosing the software that allows pens to work with computers.

Getting everyday software programs to use pen input isn't a simple matter. You can't just plug a pen add-on into your computer and expect to scribble away. In fact, using pens requires sophisticated software that goes beyond the built-in abilities of DOS.

PEN WINDOWS' BASIC COMMANDS

ACTION	COMMAND	ACTION	COMMAND
Edit Text		Copy	
Backspace		Paste	
Insert Space		Delete	
Insert New Line		Delete Word	
Cut		Extend Selection	
Undo		Insert Tab	

Microsoft Windows for Pen Computing uses simple pen strokes to perform standard computer functions such as deleting, inserting, and moving. It is included with each of the five pen attachments included in the Product Information box.

Three main contenders are seeking to become the standard interface between pen computers and application software. The first is Microsoft's Windows for Pen Computing (Pen Windows), essentially an add-on to the Windows operating environment. As it does with all its products, Microsoft is promoting Pen Windows hard. Windows and pens are made for each other, because they're both oriented to finely-grained graphics displays instead of the coarser character-based DOS screens.

GO Corp.'s PenPoint, another major player in the pen software market, is unique because it's a full-fledged, stand-alone operating system designed just for pen-based systems. (This contrasts with Pen Windows, which runs "on top" of Windows, which runs "on top" of DOS.) But there's a big "gotcha": PenPoint currently doesn't run with low-cost add-on products, but that may change in the future. Stay tuned.

Finally, PenDOS from Communication Intelligence Corp. sounds the most interesting, because it's designed to make standard DOS applications work with pens. But the applications must be modified by the program's developer; and so far only special-purpose business applications work with PenDOS. (That may also change, however, because IBM recently licensed PenDOS.)

You can't go out and buy any of these software packages—they're bundled with hardware. And so far, Pen Windows seems to be the "winner." The five pen-based peripherals that I examined come with it. (See the table, "Pen Windows' Basic Commands," to get an idea of how the program handles pen-based editing.)

Add-On Alternatives

Although all the products include Pen Windows, differences among the devices do exist. PenMate from Mouse Systems is based on older technology, employing a digitizer pad of the type used for years for tracing drawings into CAD programs. A 12¾- by 9½-inch pad (with a 9- by 6-inch active area) plugs into your computer's serial port, and a pen plugs into the pad. The included Pen Windows package comes with a few sample applications—including a word processor and sample form—that let you explore working with PenMate, which sells for $349.

Other add-on pen products take similar tacks. AceCAD's AceCAT ($179) is a smaller pad, with a 5- by 5-inch active area. The unit's size lets it rest more comfortably on an arm or knee. Appoint's Gliffic Plus ($299) looks similar, but one thing working for it is that it's cordless, using an infrared link (like your television's remote control) to communicate with your computer. Even its pen is cordless (unfortunately, making it easier to misplace.) Inforite's MP100 ($295) is the smallest of the lot and the easiest to handle, with a tiny 2½-by-4½ active area. Because Inforite also makes the computers that UPS uses, it's not surprising that the MP100 is used mainly for capturing signatures or other minor applications.

And finally, for something completely different, Arthur Dent Associates has a product called WriteAway, a pen pad that clips directly over the screen of a Zeos notebook computer. (A version for Compaq's notebook computers should be available by the time you read this.) At $595, it's the most expensive of the group, but it provides the features of a full-fledged pen computer.

All of these pen-based add-ons provide a unique ability, whether or not you use Pen Windows: You can use them as mouse substitutes. The more I used this feature, the more I liked it. Instead of dragging a mouse pointer across a screen, you place the pen anywhere in the active area, and the cursor immediately appears at that position.

These pen-based add-ons provide a unique ability: You can use them as mouse substitutes.

The jury is still out on the future of pen computing, though it seems clear that the electronic stylus will never completely replace the old-fashioned ink pen. I found using these pen-based add-on systems initially frustrating, because they're a uniquely different way of interacting with your PC. But they grow on you. If you think of them as a graphics-oriented extension of the keyboard, you can quickly start to see how useful they can be. ■

Most recently a senior editor at BYTE Magazine, *Stan Miastkowski is a free-lance writer who specializes in translating technology into English. He appears regularly on the PCTV-Live! syndicated television show.*

PRODUCT INFORMATION

AceCAT, $179
AceCAD Inc.
P.O. Box 431
Monterey, CA 93942-0431
800-676-4223
408-655-1900

Gliffic Plus, $299
Appoint Inc.
4473 Willow Road, Suite 110
Pleasanton, CA 94566
800-448-1184
510-463-3003

MP100, $295
Inforite Corp.
1670 S. Amphlett Blvd., Suite 100
San Mateo, CA 94402
800-366-4635
415-571-8766

PenMate, $349
Mouse Systems Corp.
47505 Seabridge Drive
Fremont, CA 94538
510-656-1117

WriteAway, $595
Arthur Dent Associates
500 Clark Road
Tewksbury, MA 01876
508-858-3742

Doing Windows Without Windows

Two years ago, Microsoft wowed the PC industry by introducing Visual Basic, a tool that lets you create Windows programs by using pull-down menus and the old familiar sounds of the Basic language. Now, in a departure from the usual DOS-to-Windows migration, Microsoft is selling an upgrade that brings the same easy automation to the world of DOS programming.

Visual Basic for MS-DOS is truly a new programming language, light-years ahead of GW-Basic and its recent successor, QBasic. It uses the same programming philosophy behind so-called graphical user interfaces such as Windows, OS/2, and system software for the Macintosh. Prominent among its many features—too numerous to list here—is its ability to convert QBasic and GW-Basic programs to executable (EXE) format so they can run outside QBasic's environment as grown-up programs do.

Visual Basic for MS-DOS is, in my humble opinion, clearly the direction in which serious DOS programming is moving. To show you why, I'll discuss three characteristics of the language system that illustrate how to use it: the programming paradigm, the programming environment, and the programming process itself.

The Programming Paradigm

A programming language's *paradigm* describes the way in which programmers use the language to solve computing problems. GW-Basic and QuickBasic both use a top-down, procedural paradigm. In procedural programming, you, the programmer, determine the program flow at all times, moving from one task to the next in a predefined order. The software prompts you throughout the process, perhaps asking which file to load or what operation to perform.

Visual Basic, on the other hand, uses an event-driven paradigm, under which you, the programmer, build an operating environment that allows the user to decide what's going to happen and when.

The Programming Environment

Most graphical environments such as the Macintosh interface or Windows are based on an event-driven paradigm (you click on something and something else happens). Their ease of use is the result of powerful user-interface building tools, which take care of the drudgery of monitoring the interface, letting the user move windows around the screen, resizing or relocating objects, and so on. The days of writing procedures to draw boxes on the screen and then checking for mouse clicks are over.

> **Visual Basic for MS-DOS is clearly the direction in which DOS programming is moving.**

BY ALAN L. ZEICHICK

The Program Writing Process

To prepare for this article, I wrote an event-driven program called CALC.EXE, which actually is an over-simplified pocket calculator program. With Visual Basic, you use one tool called the Form Designer to create the user interface, the physical part of the calculator. You use another tool called the Programming Environment to create the "software circuits," the parts that make the calculator function.

After launching Visual Basic, I used the mouse and pull-down menus to open a new form, and then drew on screen the objects composing the calculator, including the numeric display field and buttons. Visual Basic calls numeric display fields text boxes and refers to buttons as command buttons. (Other object types include disk-file lists, "check boxes" for yes/no choices, "radio buttons" for choosing between several options, and scroll bars.)

Next, I assigned each object appropriate "properties." In Visual Basic, an object has several properties. All objects have a "CtlName" property, which should be set to a meaningful value (such as Multiply for the multiply button). Buttons have a "caption," which tells Visual Basic what to display on the button's face. Text boxes have a Text value in which Visual Basic places some combination of characters or numbers (in this case, the numbers that the calculator works with).

Thanks to Visual Basic's built-in features, it takes only minutes to add drop-down menus to a program.

You must set some properties from the Form Designer while writing the program, but most may be changed by the program. Refer to the Photo to see how Form Designer looks after control objects are inserted.

Wiring Diagram

After I designed the program's interface, I wrote the Basic-language code that makes everything happen. Don't worry—it's a lot simpler than designing the integrated circuits inside a pocket calculator! In fact, I only wrote a few lines of Basic code. That's one of the real benefits of Visual Basic—so much is taken care of by the programming environment.

For example, to "wire" the Add push-button object, I simply double-clicked the mouse on the command button, and Visual Basic asked if I wanted to exit to the programming environment. I answered yes, and was presented with a list of all the screen objects I created, along with their associated "events," such as click, double-click, and so on. I wanted to write the program code that runs when the Add object is clicked, so I highlighted Add in the objects column, Click in the events column, and pressed the Edit in Active option to open a Basic programming window.

Visual Basic showed me two lines of code:

```
SUB Add_Click()
END SUB
```

It's an ordinary Basic subprogram, just like the ones you see in procedural languages.

To close the calculator, and return to the programming environment, I typed Ctrl-F4. After typing just one line of Basic code, I had a working adding machine! To enter the calculation line for the subtract, multiply, and divide lines, I changed only the math operator. I then had a four-function calculator, having written only a few lines of Visual Basic programming code.

To program the Done button that ends the program, I wrote:

```
SUB Done_Click ()
END
END Sub
```

The End command, without any parameters, ends the program, returning you to the programming environment (or, if you've compiled the program, sending you back to DOS).

A La Carte Menus

What user-centric program is complete without drop-down menus? Thanks to Visual Basic's near-complete list of built-in features, it took only a few minutes to add CALC.EXE's drop-down menus complete with control-key shortcuts for keyboard-only users. (Visual Basic can do everything without a mouse, although using one saves time.)

In Visual Basic's Menu Designer window, I created one main-menu object, Done, which

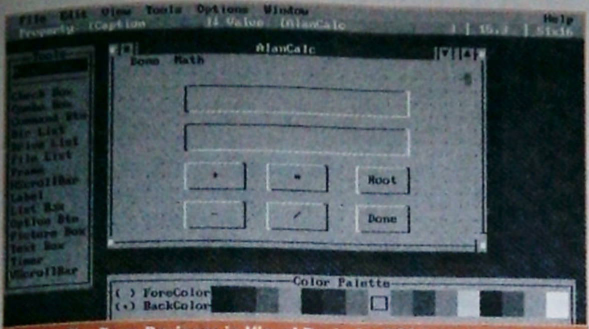


Photo. The Form Designer in Visual Basic for DOS lets you design a program's screen by choosing standardized "objects" from pull-down menus.

ends the program, and a second object, Math, which drops down to show a list of math functions: Add, Subtract, Multiply, Divide, and Root. Each has an associated control-key combination. For programming purposes, each menu item is treated as a new screen object that can be clicked.

In my calculator program, I wanted each menu item to simulate the clicking of one of the calculator's push buttons. For example, the Menu_Multiply object, when clicked, should in turn click the regular Multiply button. Its program code looks like this:

```
SUB Menu_Multiply_Click
Multiply_Click
END SUB
```

The other menu controls can be wired in a similar fashion.

Code To Go

So, how many lines of Basic code did I write? Not many. But what matters more, Visual Basic provides a way of delivering ready-to-run versions of your programs to your friends, coworkers, or customers. With the old GW-Basic, you could deliver the Basic source program, because a version of GW-Basic shipped with every DOS from 2.x to 4.x. QuickBasic added the ability to compile your program: that is, translate it into a stand-alone machine-language program that anyone could run.

Visual Basic offers that same capability: Pull down the Run menu, select Make EXE (for *make executable*), and you'll end up with a DOS file sporting an EXE extension. Compiling your code in this way gives Basic owners who don't own Visual Basic a means of running your Visual Basic programs.

Wrapping It Up

Many other Visual Basic features deserve mention. There's the ability to share Visual Basic programs with Visual Basic for Windows (called VB-Win, rather than VB-DOS). There's the sophisticated source-code debugger, helpful when your program just doesn't work right. There's QuickBasic and QBasic compatibility: You can use the editor, debugger, and compiler to work with existing QuickBasic programs, or to build new ones.

If you're already an experienced programmer, you might want to check out the Visual Basic for MS-DOS Professional Edition. Although more expensive ($495, rather than $199), it includes features I consider essential for writing business applications (but not for learning how to program): a database engine, charting and presentation graphics, financial functions, a high-speed math library, code optimization for 386 and 486PCs, a source-code profiler, and a custom Help toolkit.

Whether you choose the regular or professional edition, you have nothing to lose and everything to gain. Both support QuickBasic, QBasic, and GW-Basic, but more important, you'll get a head start on learning how to program under an event-driven paradigm, the way that software is developed for today's major graphical user interfaces on PCs, the Macintosh, and NextStep. Go for it. ■

Alan L. Zeichick is the editor-in-chief of CADENCE Magazine, *published by Miller Freeman Inc. in San Francisco.*

PRODUCT INFORMATION

Visual Basic for MS-DOS, $199
(Professional Edition, $495)
Microsoft Corp.
16011 N.E. 36th Way
Redmond, WA 98073
206-882-8080

Q&A DOS, DISKS, AND DILEMMAS

BATCH FILES

Q I am writing a batch file that will be used by operators who have little or no computer experience. Can you recommend a way to have the batch file check to see that a floppy disk is formatted before proceeding with an operation?

A If you want to use only DOS commands in your batch file, I think you are out of luck. You could use VOL to check whether the disk has a label. (Under DOS 4.x and later, a formatted disk has a label; an unformatted disk does not.) This solution might not work for novice computer users, however, because DOS displays the infamous and confusing "Abort, Retry, Fail?" message when it fails to find a volume label.

A better way to check for a formatted disk involves using a specialized batch file assistance program. You'll find many programs of this type on bulletin boards and shareware disks. One such program, which I found on CompuServe, is called CHKDRV. It checks the drive you specify and returns an Errorlevel code to indicate whether the drive and disk are ready.

—H.B.

DOS

Q When I try to format a floppy disk with the /S switch, DOS responds with the following message: "Insert DOS disk in drive A." When I insert one of my DOS disks in that drive, the same message reappears, and nothing happens. What's wrong?

A DOS is asking for a copy of the three files that it needs to create a boot disk. Two are hidden system files that generally have names such as IO.SYS and MSDOS.SYS. The third file is COMMAND.COM, which displays the DOS prompt and reacts to the instructions you type. To comply with your request to produce a bootable system disk, DOS must put those three files on the disk you are formatting.

DOS looks for the two hidden files in the root directory of the current drive; it looks for COMMAND.COM in the same location or in the directory specified by your COMSPEC environment setting. If you have partitioned your hard disk into two or more logical drives, make sure you are in the bootup partition (usually drive C) before trying to create a system disk. If you have a system with two floppy drives and designate one of them as the current drive before invoking the `FORMAT /S` command on the other floppy drive, make sure that the first drive contains a bootup disk created with the same version of DOS your hard disk uses.

—H.B.

Q I own two computers: a laptop that runs DOS 5.0 and an older 80286 PC that has DOS 3.3 installed. I use the older machine as a backup system when the laptop has problems. Although I can back up the data on the laptop using the DOS BACKUP command, when I attempt to restore the files on the 286 machine, I receive an "Incorrect DOS version" error message. What can I do?

A You're having problems because the programs controlling BACKUP and RESTORE commands have changed with each new version of DOS. Normally, files created by one DOS version's BACKUP command can't be restored using a version of RESTORE belonging to a different version of DOS.

You can create a DOS 5.0 boot disk for your 80286 computer and add RESTORE to it so that it will restore your files to the 80286's hard drive. You'll be better off, however, if you simply upgrade your desktop computer to DOS 5.0. The procedures generally are easy and painless.

An alternative, if the computers frequently are in the same office, is investing in a program, such as Traveling Software's LapLink, that can transfer files from one computer to another. Moving or backing up files with such programs is quicker than using floppy disks. File-transfer programs use either the serial or parallel ports of both computers and can move files from one machine to another at speeds of more than 500,000 bits per second.

—H.B.

WINDOWS

Q I recently installed Windows 3.1 and frequently use it to run DOS programs.

••• BY HARDIN BROTHERS AND JOHN WOLFSKILL •••

I've had no problems getting the DOS applications to run in a Windows 3.1 "window." But, my mouse goes dead when I try to use it inside the DOS application. Why?

A First, make sure you are loading the appropriate DOS driver for your mouse before you start Windows. You typically do this by adding a line such as the following to your CONFIG.SYS file:

```
DEVICE=MOUSE.SYS
```

or by invoking MOUSE.COM from your AUTOEXEC.BAT file. Keep in mind, however, that if you can't use a mouse with a particular DOS application before you start Windows, you won't be able to use it when you run that application inside Windows.

If you're using a mouse-compatible program whose driver is correctly installed but still can't use the mouse in a DOS application launched from Windows, you probably have an old version of the mouse software. Call the device's manufacturer and ask for its latest drivers. If you are using a Microsoft mouse, for example, you should be using Version 8.2 of the driver software.

—H.B.

HARDWARE

Q Sometimes my 80386 PC displays a "Memory parity error" message, provides a four-digit number telling me the segment where the problem lies, and then dies. The number is different each time, and the problem seems to occur only after several hours of continuous use? What's happening?

A Memory-parity errors such as the one you describe often are caused by a faulty memory chip or the failure of a logic chip within your computer's memory circuitry. With an intermittent problem like the one you describe, isolating the bad chip can be a difficult problem.

Your computer may be affected by "temperature bombing," a rare problem that arises when the temperature in a circuit exceeds the limit at which its chips are designed to operate.

I suggest you start by treating your computer to a thorough housecleaning. Make sure all case vents are free of dust, dirt, and debris. This will let the maximum amount of cooling air pass through the computer. Also, check the power supply fan to make sure it's running at full speed. If it sounds sluggish or is making unusual noises, try putting in a new fan to see if that corrects the problem.

Also, put your hand on or near the area of the case covering the power supply. An excessively hot case is one sign that your computer's power supply can't properly regulate its output voltage and current. This problem can raise havoc with the electronic circuits the power supply is designed to serve.

Next, open the case and "eyeball" your computer's memory chips. Look carefully for telltale signs of overheating. A chip that is hot to the touch or that contains a buildup of grayish powder on its surface is a likely candidate for replacement.

If all seems well, I suggest you purchase a utility program to test your RAM chips. While such programs are not always 100-percent effective, they can spot a bad RAM chip if it happens to fail during an extensive overnight test. If that doesn't solve your problem, take your PC to a specialist for diagnosis and repair.

—J.W.

Q The slide presentation processing service I use requires printer output in the form of a PostScript printer file. Unfortunately, the company's software isn't compatible with my DOS application. Is it possible to direct the output of my PC's parallel port, LPT1, to a disk file using a DOS command?

A DOS does not include a built-in command for redirecting the output of your parallel port to a file. You can, however, download J.M. Allen Creations' shareware program, PRINDIR 8.02, from CompuServe's IBMSYS forum (Library 3). PRINDIR is a memory-resident program that lets you send printer output to any COM or LPT port, or to a disk file.

—J.W.

Q I've heard that you sometimes can cure hard disk problems by running something called a low-level format program. What exactly do programs like these do? Where can I get one?

A A low-level format (also called a manufacturer's format) normally is performed by a drive's manufacturer before the unit is shipped to your dealer. This procedure establishes the initial address marks that divide the surface of the drive's disk platters into concentric tracks, each of which contains a number of smaller sectors.

Low-level format programs also perform a rigorous test, called a surface scan, to look for surface defects that may cause unreliable drive operation. When the program encounters a bad spot, it stores the track and sector location in a special file that later becomes the disk's bad track list. (Usually, you'll find a printed copy of this list attached to the top of your hard drive.) The list serves as a road map for identifying flawed areas of the disk.

During disk read/write operations, your hard disk controller uses the address marks established by the low-level format program to move the read/write heads from track to track. Some hard drives, especially older models, may suffer from a problem called head drift. Head drift causes the read/write heads to become misaligned with the original address marks; because the hard drive controller no longer can find the address marks, it assumes they don't exist. The result usually is a series of intermittent disk errors. When such problems arise, you can perform another low-level format to write new address marks that correctly align with the drive's read/write heads. Many low-level formatting programs let you enter the original bad track list before you begin reformatting.

Some hard drive manufacturers provide the low-level format program on a floppy disk; others make it available only to dealers. But, the program usually is stored in the hard disk controller's built-in ROM BIOS chip set. In many cases, you can access the low-level format program via DOS's Debug utility. If you can't locate a copy of your drive manufacturer's low-level format program on disk, call your dealer to find out if there's an easy way to use Debug to access the program supplied with your hard disk controller.

Low-level formatting erases all the information stored on your hard disk, so be certain to back up all your existing programs and data before you start. After you run the low-level format program, you'll also need to run DOS's FDISK.EXE (FDISK.COM in DOS 4.x and earlier) partitioning program and FORMAT.COM before restoring your original programs and data to the hard disk.

Finally, you should never perform a low-level format on a drive that follows the Integrated Drive Electronics (IDE) standard. The drive will be ruined if you do.

—J.W.

Q I recently purchased a used PC equipped with a 40MB hard drive and DOS 3.2. The drive is partitioned with a 4MB DOS partition. The remaining space is set up as a non-DOS partition. Why doesn't the FDISK program supplied with DOS 3.2 let me increase the space available for the DOS partition?

A Your older hard drive probably was partitioned using a third-party utility, such as Seagate's Disk Manager. Because DOS's FDISK program can read and alter only partitions of the type it creates, you must perform a low-level format to erase the existing data partitions and then use FDISK to re-create data partitions.

Instead of using DOS 3.2's FDISK.COM, I suggest you upgrade to DOS 5.0. It lets you use all of the space on your hard drive as one partition. Creating one partition can save you space and aggravation (see "Divvying Up Your Hard Drive," *DRG*#7, page 35).

—J.W.

HOW TO USE THIS CARD

Our Reader Service Card helps you acquire more information about software and hardware products advertised in *DOS Resource Guide.*

Subject to the request of the advertiser, each advertisement that appears in *DOS Resource Guide* has a Reader Service Number (RS#) associated with it. To receive more information about a product:

1. **Note the number that appears in the advertisement.**
2. **Circle the corresponding number on our postpaid card, opposite.**
3. **Be sure to PRINT your name and address on the card, and mail the card today.**

DOS Resource Guide

Jeff DeTray, President/Publisher
(603) 924-0249

Maureen Evans, Nat. Accounts Mgr.
(603) 924-0269 or (800) 441-4403

Linda Guyette, Sales Representative
(603)924-0130 or (800) 441-4403

Advertiser	RS#	Page
A. & H.D. Enterprises	141	76
Accessible Software	135	77
Amerifax Services Corporation	121	76
DevWare Direct	111	78-79
DevWare Direct	112	77
Diplomat Software	128	76
Direct Disk Marketing	133	76
IDG Books Worldwide	104	80
MicroLogic Software, Inc.	108	Cov2
Pendragon Software Library	115	77
Quantum Technologies	138	77
Smartluck Computer Systems	125	77
Softlogic Solutions	105	Cov4
Sunar Systems	118	76
Walnut Creek CD-ROM	130	76

SUBSCRIBE TODAY

DOS RESOURCE GUIDE

800-238-6580

SAVE UP TO 25% OFF

THE COVER PRICE

DOS RESOURCE GUIDE

READER SERVICE CARD

March 1993
Expires April 30,1993

NAME/TITLE ____________________
COMPANY ____________________
ADDRESS ____________________
CITY ____________________
STATE ____________________
ZIP ____________________
TELEPHONE ____________________
FAX NUMBER ____________________

100	110	120	130	140
101	111	121	131	141
102	112	122	132	142
103	113	123	133	143
104	114	124	134	144
105	115	124	135	145
106	116	126	136	146
107	117	127	137	147
108	118	128	138	148
109	119	129	139	149

ID8RSNC1

Are you a Subscriber? ☐ **YES** (Circle #100) ☐ **NO** (Circle #101)

Is this your copy of DOS Resource Guide.. ☐ **YES** (Circle #102) ☐ **NO** (Circle #103)

DOS RESOURCE GUIDE

READER SERVICE CARD

March 1993
Expires April 30,1993

NAME/TITLE ____________________
COMPANY ____________________
ADDRESS ____________________
CITY ____________________
STATE ____________________
ZIP ____________________
TELEPHONE ____________________
FAX NUMBER ____________________

100 110 120 130 140
101 111 121 131 141
102 112 122 132 142
103 113 123 133 143
104 114 124 134 144
105 115 124 135 145
106 116 126 136 146
107 117 127 137 147
108 118 128 138 148
109 119 129 139 149

ID8RSNC2

Are you a Subscriber? ☐ YES (Circle #100) ☐ NO (Circle #101)

Is this your copy of DOS Resource Guide.. ☐ YES (Circle #102) ☐ NO (Circle #103)

YOU Be the Editor!

Help us plan future issues of DOS Resource Guide! We need your opinions about our editorial coverage. Please fill out and mail this card today!

1. How much coverage of new products do you want to see in DOS Resource Guide? More reviews of new hardware and software will mean less space available for DOS hints, tips, and tutorials. What do you think? **Check one only:**
   - ☐ Don't change a thing! Keep the mix just like it is.
   - ☐ More hardware reviews
   - ☐ More software reviews
   - ☐ More hardware AND software reviews

2. Which new product categories interest you the most? **Check up to four (4):**
   - ☐ Business/Finance
   - ☐ Games
   - ☐ Word Processing
   - ☐ Communications
   - ☐ Graphics
   - ☐ Database
   - ☐ Programming
   - ☐ Education
   - ☐ Spreadsheet
   - ☐ Engineering
   - ☐ Utilities

3. How much will you spend on DOS software in the next 12 months?
   - ☐ Less than $100
   - ☐ $100-$300
   - ☐ $300-$600
   - ☐ $600-$1,000
   - ☐ More than $1,000

4. How much will you spend on computer hardware in the next 12 months?
   - ☐ Less than $200
   - ☐ $200-$500
   - ☐ $500-$1,000
   - ☐ $1,000-$2,000
   - ☐ More than $2,000

5. How many people read your copy of DOS Resource Guide?
   - ☐ One
   - ☐ Three
   - ☐ Two
   - ☐ Four or more

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 1007 PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES

DOS RESOURCE GUIDE
Attn: Reader Service Department
PO Box 5384
Pittsfield MA 01203-9733

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 1007 PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES

DOS RESOURCE GUIDE
Attn: Reader Service Department
PO Box 5384
Pittsfield MA 01203-9733

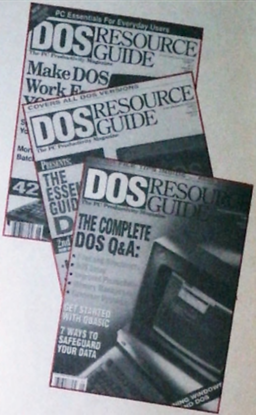
PLACE
STAMP
HERE

DOS RESOURCE GUIDE
80 Elm Street
Peterborough NH 03458

SUBSCRIBE TODAY

DOS RESOURCE GUIDE

800-238-6580



SAVE UP TO 25% OFF

THE COVER PRICE

ARTICLE INDEX 1—7 DRG

AUTHOR	ARTICLE TITLE	ISSUE	PAGE	DESCRIPTION
DOS EXPLAINED				
Brothers	"ANSI.SYS: A DOS Treasure"	*DRG#3*	48	This trove of DOS functions gives you greater control over your keyboard and display.
Brothers	"How to Resolve Conflicts with DOS"	*DRG#3*	26	Understanding DOS error messages helps you set problems straight.
De Pasquale	"Command Performance"	*DRG#6*	41	Use ANSI.SYS to reprogram your keyboard's function keys.
DeVoney	"DOS Magic Tricks: Piping and I/O Redirection"	*DRG#3*	44	Mastering DOS's redirection symbols saves you time and keystrokes.
Gookin	"10 Worst DOS Commands"	*DRG#5*	23	One computerist's exposé on the dogs of DOS.
Nelson	From the Bookshelf: "DOSKEY Revealed"	*DRG#5*	28	Tips and tricks for putting DOS 5.0's DOSKEY command to good use.
Nimersheim	"DOS Goes Graphical"	*DRG#4*	15	Getting to know DOS 5.0's graphical operating environment.
Nimersheim	"Give DOS 5.0 a Friendlier Face"	*DRG#6*	17	What you can and can't do to personalize DOS 5.0's initialization file.
Nimersheim	"Upgrade Now!"	*DRG#5*	40	A half-dozen reasons why you should switch to DOS 5.0.
Rowell, Wolfskill	"Guide to DOS Files"	*DRG#3*	14	Find out in an instant what all the files in your DOS subdirectory do.
Veale	"The DOS Version That's Right for You"	*DRG#1*	8	Find out if you have the best DOS version for your needs. (Does not cover DOS 5.0.)
Wolfskill	"DOS Shortcuts"	*DRG#1*	50	Tricks to save time and keystrokes when you work at the DOS prompt.
Wolfskill	"A Farewell to Incompatibility"	*DRG#5*	19	APPEND.EXE, ASSIGN.EXE, DRIVER.SYS, and SETVER.EXE can bring peace to warring software and hardware.
Wolfskill	"Inside DOS 5.0"	*DRG#3*	4	Find out what this "significant" DOS upgrade has to offer.
Wolfskill	"A Six-Pack of DOS Treasures"	*DRG#4*	29	RAMDRIVE.SYS, MODE.COM, ANSI.SYS, GRAPHICS.COM, SMARTDRV.SYS, and DOSKEY.COM are utilities you easily can learn to love.
FILES, DIRECTORIES, AND HARD DRIVES				
Badgett	"Divvying Up Your Hard Drive"	*DRG#7*	35	When you should and should not consider partitioning your hard drive.
Brothers	"Conjure a Solution to Drive Dilemmas"	*DRG#3*	75	ASSIGN, JOIN, SUBST, and APPEND can trick DOS into seeing drive organization your way.
Brothers	"Hard Disk Tactics"	*DRG#1*	20	Proven strategies to help you untangle an unruly hard disk.
Brothers	"XCOPY Express"	*DRG#3*	34	XCOPY does almost everything COPY does, and it does it better, faster, and with less fuss.
Langdell	"Of Files and Directories"	*DRG#4*	9	An introduction to the DOS commands that let you create, store, recall, and transform files.
Langdell	"Positive Attributes"	*DRG#5*	15	Discover the ways ATTRIB can help you work more efficiently with files.
Norton, Ashley, and Fernandez	From the Bookshelf: "Advanced Class in Copying Files"	*DRG#6*	26	Learn the finer points of copying files with XCOPY, COPY, and REPLACE.
Rosenfelder	"CHKDSK to the Rescue"	*DRG#3*	37	When disaster strikes your files, CHKDSK helps salvage important data.
Rosenfelder	"COPY: New Tricks for an Old Command"	*DRG#3*	30	Using COPY to retrieve portions of an unreadable file, check a disk, and view the contents of a file.
Rowell	"Back Up to Be Sure"	*DRG#1*	40	The BACKUP and RESTORE commands will serve you well for occasional big backup jobs.
Rowell	"Organize Your Hard Drive"	*DRG#3*	72	Creating an efficient subdirectory system simplifies computing sessions.
Wolfskill	"The Importance of Good Timing"	*DRG#1*	54	Knowing how and when to use CHKDSK can save you time and effort when reconstructing damaged files.
SETUP AND CONFIGURATION				
Brothers	"Develop Your PC's Personality"	*DRG#5*	11	Fine-tuning AUTOEXEC.BAT and CONFIG.SYS is a good way to get your computer to behave the way you think it should.
Brothers	"Managing DOS Setup Files"	*DRG#4*	19	Getting acquainted with AUTOEXEC.BAT and CONFIG.SYS can help you prepare for inevitable bootup problems.
Houston	"Boot Disks for Every Occasion"	*DRG#1*	4	Build a library of boot disks that customize your CONFIG.SYS and AUTOEXEC.BAT files for every application.
Rowell	"Start-Up Files for a Perfect Fit"	*DRG#3*	23	Tailoring CONFIG.SYS and AUTOEXEC.BAT to your needs.
MEMORY MANAGEMENT				
Alford	"Get Your Memory Straight"	*DRG#1*	64	Sort through the alphabet soup of memory acronyms and find out what every PC user should know about ROM, EPROMs, SRAM, DRAM, and VRAM.
Glass	"Sorting Out Memory"	*DRG#4*	38	Understanding the differences between expanded and extended memory is crucial to successful computing.
Langdell	"A Memory Glossary"	*DRG#1*	66	A glossary to help you master the ins and out of memory manipulation.
Langdell	"The Roots of DOS"	*DRG#7*	17	DOS won't seem quite so confusing if you understand how it evolved from earlier operating systems, such as CP/M and Unix.
Wolfskill	"Memory Management Made Easier"	*DRG#3*	10	Simple steps to putting DOS 5.0's program relocation schemes to work in the lower 640K.
BATCH FILES				
Axelson and Hughes	"Make Your Batch Files Stop, Look, and Listen"	*DRG#3*	65	Expand your computing repertoire with batch files that pause and ask for directions.

Continued on Page 74

ARTICLE INDEX 1—7

Continued from Page 73

AUTHOR	ARTICLE TITLE	ISSUE	PAGE	DESCRIPTION
		BATCH FILES (Continued)		
Bee	"List 'Em by the Batch"	*DRG#1*	14	This handy batch file helps you manage your appointments, records, and short lists.
Bee	"Take Control with Batch Files"	*DRG#4*	55	Simple hints for using batch files to take control of your computing sessions.
Brothers	Part 1: "Programming in DOS's Batch Language"	*DRG#6*	37	To write a batch file, start by mastering REM, PAUSE, ECHO, and CALL.
Brothers	Part 2: "Advanced Batch Files? No Problem!"	*DRG#7*	31	Improve your batch files with SHIFT, GOTO, IF ==, IF ERRORLEVEL, IF EXIST, FOR...IN...DO, line labels, and command-line parameters.
Broussard	"Backup by .BAT"	*DRG#1*	44	Use BACKER.BAT to back up only the files you need to save.
Burnett	"Box It Up"	*DRG#5*	63	Two short batch files turn your keyboard into a tool for designing boxes.
Holtzman	"Batch File Basics"	*DRG#3*	54	Learn how to create batch files and reduce the tedium of performing repetitive computing tasks.
Nebel	"Hard Disk Housekeeping"	*DRG#5*	65	Batch files to help you keep your subdirectory system neat.
Rowell	"Raise Your Batch File IQ"	*DRG#3*	61	Add brainpower to your batch files by tapping the resources of the DOS environment space.
Wolfskill	"Unlocking the Secrets of Exit Codes"	*DRG#7*	20	To use an ERRORLEVEL statement effectively in batch files, you need to know how to make use of exit codes.
Simpson	"Once a Day"	*DRG#1*	18	This batch file runs a program the first time your computer is turned on or rebooted each day.
		PROGRAMS AND PROGRAMMING		
Bannon	"Building a QBasic Database"	*DRG#6*	65	Keep track of names and addresses with this easy-to-use QBasic program.
Brothers	"Elementary Steps to Programming"	*DRG#4*	63	Build a phone list utility while learning to program in QBasic.
Brothers	"An Express Ramp to Your Subdirectories"	*DRG#3*	78	GO.EXE takes you from subdirectory to subdirectory—fast.
Gookin	Part 1: "Introducing Debug"	*DRG#5*	34	Overcome your fear of DOS's Debug utility by trying these experiments.
Gookin	Part 2: "Debug Does Disks"	*DRG#6*	21	How to make Debug read to and from disk—safely.
Gookin	Part 3: "Write Your Own Debug Utilities"	*DRG#7*	25	Learn to write Debug scripts and turn them into programs you can execute at the DOS prompt.
Sessions	"Seven Programs to Safeguard Your Files"	*DRG#4*	69	Use these batch files, Debug scripts, and Basic programs to protect data from loss or theft.
Wolfskill	Part 1: "Programming from A to Z"	*DRG#5*	56	Explore the "anatomy" of QBasic programming while typing in this printer-control utility.
Wolfskill	Part 2: "The Importance of Good Thinking"	*DRG#6*	61	Plot your course before you set out to write a QBasic program.
Wolfskill	Part 3: "Building a Solid Foundation"	*DRG#7*	57	Creating an initialization module sets up your QBasic program for what follows.
Zeichick	"The Big Clock"	*DRG#7*	47	Face the challenge of solving trigonometric problems while creating a QBasic program that displays a large analog clock on your screen.
		HARDWARE		
Alford	"10 Hints for a Healthy Hard Drive"	*DRG#1*	26	10 ways to give your hard drive a long and healthy life.
Bee	"The PC User's Survival Kit"	*DRG#1*	36	How to hex the hardware gremlins before they zap your computer.
Brothers	"Flip the Switch"	*DRG#1*	60	What happens inside your computer when you turn on the switch.
Glass	"Understanding Serial Ports"	*DRG#5*	44	How to avoid potential serial port problems.
Mosteller	"Format Your Hard Drive for Speed"	*DRG#3*	70	How to improve the performance of older AT-type hard drives.
Veale	"Boosting Your PC's Memory"	*DRG#4*	51	Step-by-step instructions for choosing and installing more memory.
Veale	"The Seven Deadly Sins of Computing"	*DRG#1*	62	Advice on dealing with the seven biggest threats to the safety of your hardware and software.
Veale	"Speed Up Your Hard Drive"	*DRG#1*	30	Boost hard drive performance without investing in new hardware.
Wolfskill	"On the Upgrade"	*DRG#4*	45	A small investment in hardware can give an aging computer a performance boost.
		SOFTWARE		
Marshall	"More Disk Space for Free"	*DRG#6*	56	Hard disk overcrowded? File and disk compressors open up space aplenty.
Wolfskill	"DR DOS 6.0: A Most Worthy Opponent"	*DRG#6*	51	This MS-DOS alternative offers a number of good reasons for leaving Microsoft behind.
Emigh	"The Connoisseur's Shareware Library"	*DRG#7*	43	These eight low-cost shareware programs deserve a place in your software library.
		GENERAL COMPUTING HELP		
Nimersheim	"How to Beat Modem Anxiety"	*DRG#6*	45	Today's software and modems make it easy to use bulletin boards and exchange files.
Zeichick	"Protect Your Data from Computer Viruses"	*DRG#5*	50	Ward off potential problems by educating yourself about computer viruses
		WINDOWS		
Livingston	From the Bookshelf: "Using DOS Under Windows"	*DRG#6*	70	How to achieve peaceful coexistence between DOS and Windows.
Nelson	From the Bookshelf: "A Little Voodoo Can Raise Windows from the Dead"	*DRG#7*	66	35 tips that make setting up Windows less of a chore.

Compiled by Marilyn McMaster

TAKE THE DREAD OUT OF DOS!

If you're trying to succeed with DOS, but feel like a General Failure...

DOS RESOURCE GUIDE IS WRITTEN FOR YOU

DOS RESOURCE GUIDE, the bi-monthly periodical packed with tips, tutorials, and information on all aspects of PC computing, is your antidote for computing confusion.

DOS RESOURCE GUIDE #1

Make DOS Work for You

Covers six areas of computing that are vital to DOS users: start-up files (AUTOEXEC.BAT and CONFIG.SYS), batch file creation, hard drives, safe backups, organizing file management, and hardware maintenance. [illegible] List'em, a batch file that helps you [illegible] app[illegible]; [illegible] tips to keep your [illegible]; speed-up tips for a logy hard drive; a batch file that automates backups; DOS shortcuts, hardware dos and don'ts. Plus 42 provocative pointers on DOS

SOLD OUT

DOS RESOURCE GUIDE #2

Guide to Windows Software

SOLD OUT

DOS RESOURCE GUIDE #3

The Essential Guide to DOS

SOLD OUT

DOS RESOURCE GUIDE #4

The Complete DOS Q&A

Breaks DOS into functional sections: Fundamentals, Set-up, Improving Productivity, Memory Management, Hardware Upgrades, Batch Files, and Using Windows. Each section is followed by a question and answer session on the topic. Plus, seven short programs to protect your data; do-it-yourself memory, advice on CONFIG.SYS and AUTOEXEC.BAT, and beginner's Basic. Not to mention nine tips for the DOS timid.

DOS RESOURCE GUIDE #5

The One-Stop DOS Primer

All you can eat: DOSKEY, SETVER, APPEND and ASSIGN, plus a tutorial on using the ATTRIB command. Part 1 of a Debug series. Our "10 Worst" list of DOS commands. Your introduction to serial ports. Includes a Qbasic Printer Control Program: Just type and run. Sharpen your DOS finesse with 15 type-in tips.

DOS RESOURCE GUIDE #6

13 Ways to Boost Your DOS I.Q.

Peter Norton shares some valuable COPY secrets. Personalizing the DOS 5 shell. Part II: altering disk data with DEBUG. Modem communications made simple. Meet the DOS competition—our first look at DR-DOS. Compression/decompression software explained. Part I of a batch file clinic; FREE for the typing: a QBasic name and address database. Plus, the lucky 13—tips, that is—for boosting your DOS I.Q.

Back issues of DOS Resource Guide are $5.95* postpaid!

Use our 800 number and have your credit card ready!

1-800-238-6580

Or send your check or money order to:

DOS Resource Guide
P.O. Box 3162
Salisbury, MD 21802

*Maryland residents add 5% sales tax. Back issues to Canada and Mexico cost $6.95 (Postpaid). Other International (Surface) $8.00 (Air) $13.00. Checks and money orders must be payable in U.S. funds, drawn on a U.S. bank.

VISA, Mastercard, or American Express also accepted. Please include your card number, its expiration date, and your signature on your request.

Please indicate which issue(s) of DOS Resource Guide you are ordering.

DON'T MISS AN ISSUE!

DRG MARKETPLACE

READERS: To help you learn more about the many interesting products available for your PC, **DOS Resource Guide** has created the ***DRG Marketplace***. This special section makes it easy to contact companies offering a wide range of accessories, software, and services. Listings are grouped by product category for your convenience.

VENDORS: Listings in the ***DRG Marketplace*** are sold by the column inch. *For more information, contact Linda Guyette at **1-800-441-4403 or 603-924-0130.***

HARDWARE

COMMUNICATIONS

PC VOICE MAIL KIT $1795

- Voice Processing on your PC
- Software and 4 line board included
- Order Taking
- AUDIOTEX
- Voice Messaging
- 2 year warranty

Never miss another call! Hundreds of Uses!
Don't pay $1,000's
***CALL TODAY** for **FREE** details.*

Amerifax Services Corp.
572 W. Market St., #5, Akron, OH 44303
(216) 253-8787
(216) 762-6050 demo

Circle 121 on Reader Service Card

SOFTWARE

CDROM

Walnut Creek CDROM
1547 Palos Verdes Mall, Suite 260 n Walnut Creek, CA 94596
+1-800-786-9907 n +1-510-947-5996 n FAX +1-510-947-1644

Simtel20 MSDOS CDROM	**$24.95**

640 mb in 9000+ files. Programming tools, DOS utilities, tech docs, comm, bbs, publishing, ham-radio, education, much more. Dated December 1992.

CICA MS Windows CDROM	**$24.95**

Hundreds of MS Windows programs. Utilities, games, source code, programming tools. Dated November 1992.

*Shareware programs require separate payment to authors if found useful.

Source Code CDROM*	**$39.95**
X11R5 and GNU CDROM	**$39.95**
OS/2 Archive CDROM*	**$24.95**
AB20 Amiga CDROM*	**$24.95**
CDROM Caddies	**$4.95**

Circle 130 on Reader Service Card

SOFTWARE

FOREIGN LANGUAGES

Diplomat Software Series
Alphabets of the World For Your PC
We Make Alphabets

Russian
Ukrainian
Armenian
Hebrew
Greek
Czech

Vietnamese
Lithuanian
Hungarian
Polish
Arabic
and more

Series of 36 alphabets providing both English and foreign characters for screen, keyboard & printer support while using existing software: WordPerfect, Word, Windows, Quattro Pro, Ventura Publisher, Works, dbase...and more.

DOS AND WINDOWS SUPPORT

DOS Package	$95
Windows Support	$125

(PostScript & TrueType fonts)
Prices shown for each language.
VISA, MC, COD accepted.
s/h $5 (Overseas $25)

Diplomat Software
3700 Campus Drive, Suite 202, Newport Beach, CA 92660
Phone (714) 474-6968, Fax (714) 250-8117

Circle 128 on Reader Service Card

To place your ad in
DRG Marketplace
call (603) 924-0130
or (800) 441-4403

HARDWARE

ACCESSORIES

DIRT CHEAP
MULTIMEDIA SOFTWARE & HARDWARE
1-800-22SUNAR
OR
(408) 982-0288

Circle 118 on Reader Service Card

ATTENTION HARDWARE VENDORS!
Over the next 12 months, **DOS Resource Guide** readers will buy:
15,000 New PCs **10,000 Monitors** **17,000 Printers**
14,000 Hard Drives **11,000 Modems**
Shouldn't ***your*** product be advertised in **DOS Resource Guide?**
1-800-441-4403 or **603-924-0130**

MISCELLANEOUS

ON-LINE SERVICES

WELCOME TO THE 21ST CENTURY!

POWERVISION is the new on-line computer service which lets you save money with "No-Load" shopping and referrals to Local Vendors who give discounts to **POWERVISION** subscribers.
See **PHOTOS** of the products! Simple to Use! Encyclopedias, news, stocks, horoscopes, other information.

For full color brochure and subscription form call
(301) 718-AHDE

Independent Associate
POWERVISION™
Your Link to the World™

A. & H. D. Enterprises
6110 Ridgefield Rd.
Suite 108/109
Bethesda, MD 20816

Tel. (301) 718-2433
Fax (301) 718-7346
Powervision ID: PVEMBG10

Circle 141 on Reader Service Card

MISCELLANEOUS

DISKETTES

DIRECT DISK MARKETING

Due to volume purchasing power DDM is now able to offer truly affordable diskettes. All our diskettes are pre-formatted and include sleeves and user labels.

5.25"	DSDD	$3
5.25"	HD	$4.25
3.5"	DSDD	$5
3.5"	HD	$7.50

BOXED BY 10

100% SATISFACTION GUARANTEE
Duplication Services Available
TOLL-FREE
1-800-567-5311
PH (604) 942-5718 FAX (604) 942-9346

Circle 133 on Reader Service Card

Did you know...

In the next 12 months, readers of **DOS Resource Guide** will spend:

- more than **$55 million** on hardware
- **$22 million** on software

Call **NOW** for to reserve your space in the next issue of **DOS Resource Guide!**

1-800-441-4403
or **603-924-0130**

SOFTWARE

ENTERTAINMENT/GAMES

BEAT THE LOTTERY

38 Lotto Jackpot Winners WON 78.8 MILLION DOLLARS with Gail Howard's Systems!

The ONLY Lottery System with Documented Jackpot Winners.

Gail Howard's *ALL NEW* ADVANTAGE PLUS™

Use ADVANTAGE PLUS™ and you'll trash all your other lottery software. It's the most complete, fastest and easiest to use—in a class by itself. Nothing can begin to compare!

- **NO OTHER SOFTWARE HAS *MORE* SUCCESSFUL SCIENTIFIC TOOLS FOR PICKING WINNERS.**
- Includes ALL Data for 50 pick 5 and pick-6 Lotto Games **FREE.**
- Automatic SMART PICKS™ selects best Lotto Numbers instantly.
- Test Past Accuracy of SMART PICKS™ with One Key Stroke!
- Has Over 50 Scientific Charts/Reports to Zero in on Winners.
- A $295.00 Value. Your introductory price for a limited time only: $79.95 + $3.00 S/H (IBM/Comp. 5.25 or 3.5).

SMART LUCK® COMPUTER WHEEL™
$39.50 + $3 S/H
Has 252 Lotto Systems With Specific Win Guarantees. Find One System That Fails Its Win Guarantee, and You Get DOUBLE Your Money Back!

COMPUTER WHEEL *and* COMPUTER ADVANTAGE
Save $20! Buy BOTH for Only $99.95 + $3.00 S/H

VISA **GO WITH A PROVEN WINNER!**
CALL 1-800-876-4245

SMART LUCK® SOFTWARE
Dept. RS-12, P.O. Box 1519 • White Plains, NY 10602
1-800-876-GAIL (4245) or 914-761-2333

Circle 125 on Reader Service Card

SOFTWARE

PUBLIC DOMAIN/SHAREWARE

DevWare ToolChest

IBM PUBLIC DOMAIN SOFTWARE

Hundreds of the latest and hottest Public Domain and Shareware programs available for the IBM. Our extensive catalog contains only the highest quality programs available. All types of utilities, applications, educational, home, business and entertainment software is available at incredibly low prices. You wont find better software for a lower price. Call or write us today for your free, no obligation, catalog!

DEVWARE, INC.
12520 Kirkham Court
Suite 1-DRC
Poway, CA 92064
or call us toll-free at:
(800) 879-0759

Circle 112 on Reader Service Card

Software Industry's Best Kept Secrets

Cutting-edge data compression technology slashes the cost of topnotch DOS shareware to an industry-annihilating all-time low! Don't turn into another CD-ROM shareware packrat. Get the best!

QUANTUM TECHNOLOGIES
10985 Pinehigh Dr, IBM Dept.
Alpharetta, Georgia 30201

Call 24 hrs for FREE InfoPak!
1-800-677-2182

Circle 130 on Reader Service Card

FREE SHAREWARE CATALOG
"BUY 5 — CHOOSE 2 FREE!"

100's of IBM PUBLIC DOMAIN and SHAREWARE PROGRAMS!
• VIRUS FREE • LATEST VERSIONS

PHONE/FAX **800-827-8209**

Accessible Software
500 GARDEN LN., RT. 8, E. PEORIA, IL 61611

Circle 135 on Reader Service Card

Software Vendors!

DOS Resource Guide readers plan to buy more than **70,000 software applications** in the next 12 months! Call **today** for information on how you can reach these **DRG** buyers!

1-800-441-4403 or **603-924-0130**

SAY YOU SAW IT IN THE
DRG MARKETPLACE

PENDRAGON Software Library

Public Domain/Shareware for IBM & Compatibles
No Viruses * Fast Delivery * Latest Versions
ASP Member

FREE 76 page CATALOG
1-800-828-DISK

As low as $1.75 each *** MC/Visa Accepted
75 Meadowbrook Rd, East Greenwich, RI 02818
(401)884-3475 or (401)884-6825 for Inquiries - 10AM to 10:30 PM EST

Circle 115 on Reader Service Card

DevWare Direct

IN OUR SEVENTH YEAR!

PRODUCTIVITY

Act! for DOS (Power Up!)	P1200	$263
Act! for Windows (Power Up!)	P1201	$275
Automap	P1256	$65
Business Contacts (Disk-Count)	P1167	$39
DacEasy Instant Accounting	P1202	$38
DataManager (Timeworks)	P1168	$29
Desktop Set 3.0 (Okna Corp.)	P1169	$105
Eight in One (Spinnaker)	P1170	$40
Enable Worksplus	P1203	$99
Family Origins (Parsons)	M1229	$29
Financial DOMEPlus (Dome)	R1216	$59
First Publisher (Spinnaker)	P1171	$95
GeoWorks Desktop (GeoWorks)	P1172	$49
GeoWorks Pro (GeoWorks)	P1173	$135
Grammatik 5 V 1.0 for DOS (Reference Software)	P1204	$65
Grammatik V 3.0 for Windows (Reference Software)	P1300	$65
Instant Recall (Chronologic Corp)	P1174	$89
It's Legal (Parsons)	M1230	$45
LotusWorks 3.0 (Lotus)	P1301	$109
Managing Your Money 8.0 (Mega)	P1205	$57
Microsoft Excel	P1206	$305
Microsoft Word 2.0	P1207	$219
Microsoft Word 2.0 for Windows	P1215	$307
Microsoft Works (Microsoft)	P1175	$105
Microsoft Works for Windows 2.0 (microsoft)	P1302	$139
Money Counts 7.0 (Parsons Technology)	R1217	$55
Multicalc (PC Works)	P1303	$65
M. Y. O. B. 3.0 for Windows (Teleware) PC World Best Buy!	P1304	$153
Object Vision 2.1 (Borland)	P1305	$109
Newsroom Pro (Spinnaker)	P1176	$29
On Time for Windows 3.1 (Campbell Services)	P1177	$89
Pacioli 2000 (M-USA)	P1208	$39
PC File (Buttonware)	P1306	$95
PFS: WindowWorks 2.0 (Spinnaker)	P1307	$79
PFS: Write (Spinnaker)	P1179	$85
PFS: Prospect (Spinnaker)	P1209	$94
Peachtree Basic Accounting (Peachtree)	R1219	$79
Personal Portfolio Manager (Abacus)	R1220	$59
Personal R-Base 1.0 (Microrim)	P1178	$71
ProCALC 3D (Parsons Technology)	P1180	$54
Publish It 2.0 (Timeworks)	P1181	$89
Publish It Lite (Timeworks)	P1182	$42
Quattro Pro WinDOS (Borland)	P1308	$319
Quicken 6.0 (Intuit)	P1211	$51
Quicken/Turbo Tax (Intuit)	R1218	$109
Reality's Smart Investor (Reality Technologies)	P1212	$73
Resident Expert - Effective Writing (Parsons)	P1183	$39
SideKick 2.0 (Borland International)	P1185	$69
TurboTax Personal 1040 (Chipsoft)	P1309	$49
TurboTax State Tax Pkg. (Chipsoft)	P1310	$43
Word Science Instant Definitions	P1311	$65
WordPerfect 5.1	P1214	$319
WordPerfect for Windows	P1215	$319
WordWriter Pro (Timeworks)	P1186	$65

UTILITIES

Andrew Tobias (MECA)	U1187	$55
Becker Tools 3 for Windows Upgrade (Abacus)	U1254	$47
Brooklyn Bridge (Fifth Generation Syst.)	U1188	$99
Carbon Copy Plus (Microcom)	U1189	$134
Central Point Backup 7.1 (Central Point)	U1190	$99
Dashboard for Windows (Hewlett Packard)	U1210	$59
Disk Optimizer (SoftLogic Solutions)	U1191	$49
Devware Utility Disk - 15 programs	U1192	$20
DoubleDisk 2.0 (Vertisoft)	U1300	$69
Double Density (Abacus)	U1253	$59
Fastback Plus 3.04 (Fifth Generation Syst.)	U1193	$129
FastTrax 4.0 (FastTrax International)	U1301	$45
Laplink Pro 4.0 (Traveling Software)	U1302	$115
Netroom 2.0 (Helix)	U1195	$67
NoMouse for Windows (Abacus)	U1255	$39
Norton Backup (Symantec)	U1196	$89
Norton Utilities 6.01 (Symantec)	U1197	$105
OPTune 1.3 (Gazelle Systems)	U1198	$59
PC Kwik Power Disk 1.1 (Multisoft Corp.)	U1199	$56
PC Tools 7.1 (Central Point)	U1200	$125
QEMM-386 6.02 (Quarterdeck Office Syst.)	U1201	$69
QMAPS 2.1 (Quantel Corp.)	U1202	$69
Resident Expert - All DOS Versions (Parsons)	U1203	$33
Software Carousel for Windows, OS/2 (SoftLogic Solutions)	U1303	$69
Spinrite (Gibson Research)	U1204	$63
Tempest (Abacus)	U1205	$25
386MAX 6.01 (Qualitas)	U1206	$69
ViruCide Plus (Parsons Technology)	U1207	$35
Virusure (MSI)	U1208	$47

OPERATING SYSTEMS

Dr.DOS v6 (Digital Research)	R1214	$75
Microsoft QuickBASIC 4.5	R1300	$69
MS-DOS 5.0	R5214	$79
NewWave 4.0 for Windows (Hewlett Packard)	R1230	$85
Norton Desktop 2.0 for Windows (Symantec)	R1231	$125
OS/2 2.0 (IBM)	R1232	$161
Turbo C++ 3.0 (Borland)	R1301	$75
Turbo C++ 3.1 Windows (Borland)	R1302	$119
Turbo Pascal 6.0 (Borland)	R1303	$115
Windows 3.1 (MicroSoft)	R1215	$109

EDUTAINMENT

-Top 10 Edutainment Titles -

1. Where/World is Carmen Sandiego (Broderbund)	D1165	$33
2. Where/U.S.A. is Carmen Sandiego (Broderbund)	D1164	$34
3. Mavis Beacon Teaches Typing (Software Toolworks)	D1149	$34
4. New Math Blaster Plus (Davidson)	D1154	$35
5. Playroom (Broderbund)	D1156	$37
6. Where/Americas Past/Carmen Sandiego (Broderbund)	D1162	$39
7. Oregon Trail (MECC)	D1155	$33
8. Treehouse (Broderbund)	D1160	$41
9. Where in Time is Carmen Sandiego (Broderbund)	D1163	$35
10.Reader Rabbit 2 (Learning Company)	D1158	$41
An American Tail (Capstone) Ages 7+	D1200	$33
Ancient Empires (Learning Company)	D1201	$35
Annabel's Dream - Ancient Egypt (Texan Caviar)	D1141	$54
Body Illustrated (Spirit of Discovery) Ages 12+	D1202	$49
Bodyworks (Automap)	D1170	$41
Castle of Dr. Brain (Sierra) Ages 12+	D1203	$34
Chemistryworks (Automap)	D1171	$41
Designasaurus II (Britannica)	D1142	$29
Destination Mars! (Compu-Teach) Ages 11 to 17	D1204	$39
Dinosaur Designer (Hi Tech Expressions) Ages 5 to 15	D1205	$29
Facts in Action (Spirit of Discovery) Ages 6+	D1206	$34
Follow the Reader (Walt Disney) Ages 4 to 7	D1207	$34
French Assistant (Microtac)	D1208	$67
German Assistant (Microtac)	D1209	$67
Hare Raising Havoc (Disney Software)	D1143	$27
Headline Harry and Paper Race (Davidson)	D1144	$33
Island of Dr. Brain (Sierra)	D1210	$34
Katie's Farm (Broderbund)	D1145	$17
KidPix (Broderbund)	D1146	$35
KidWorks (Davidson & Associates)	D1147	$35
KidWorks II (Davidson & Associates) Ages 4 to 10	D1211	$37
Knowledge Adventure (Knowledge Adventure)	D1212	$49
Mario Teaches Typing (InterPlay)	D1213	$29
Math Maze (Britannica)	D1148	$29
Mavis Beacon Teaches Typing (Software Toolworks)	D1149	$34
Mixed Up Fairy Tales (Sierra On-Line)	D1150	$33
My Paint (Saddleback Graphics)	D1151	$35
My Paint - Alphabet Fun Coloring Disk	D1152	$15
My Paint - Majellx Coloring Disk	D1153	$15
New Math Blaster Plus (Davidson) Ages 6 to 12	D1214	$37
Number Munchers (MECC)	D1215	$33
Operation Neptune (Learning Company) Ages 8 to 14	D1216	$37
Orbits (Automap)	D1172	$35
Oregon Trail (MECC) Best Seller	D1155	$33
PC Globe V.5.0 (PC Globe)	D1157	$43
Reader Rabbit's Ready For Letters (Learning Company)	D1217	$39
Reading Adventures in OZ (Davidson) Ages 4 to 9	D1218	$35
Rodney's Fun House (Broderbund)	D1219	$33
Science Adventure (Knowledge Adventure) Ages 5+	D1220	$49
Secret Island of Dr. Quandary (MECC) Ages 8+	D1221	$39
See the U. S. A.: New Quiz Edition (Compu-Teach) Ages 8+	D1222	$35
SimLife (Maxis) Ages 10+	D1223	$44
Snoopy's Game Club (Accolade) Ages 3 to 8	D1224	$33
Spanish Assistant (Microtac)	D1225	$67
Storybook Weaver (MECC)	D1159	$33
Studyware (ACT/SAT) (Cliff Notes)	D1226	$33
Super Solvers: Outnumbered (Learning Co.) Ages 7 to 10	D1227	$39
Super Solvers: Spellbound (Learning Co.) Ages 7 to 10	D1228	$39
Time Riders in America's History (Learning Co.) Ages 10+	D1229	$39
Turbo Science (Sierra)	D1230	$34
Where in America's Past - Carmen Sandiego (Broderbund)	D1162	$39
Where in Europe - Carmen Sandiego (Broderbund)	D1161	$35
Where in U. S. A. Carmen Sandiego Deluxe (Broderbund)	D1231	$49
Where in World Carmen Sandiego Deluxe (Broderbund)	D1232	$49
Word Munchers (MECC)	D1233	$31
World Atlas (Software Toolworks)	D1234	$49
Zoo Keeper (Davidson & Associates)	D1166	$39

GRAPHICS

Animation Studio (Disney Software)	G1209	$75
Corel Draw 3.0 (Corel Systems)	G1215	$379
DeluxePaint Animation (Electronic Arts)	G1210	$95
DesignCAD 3D 4.0 for DOS (American Small Business)	G1300	$289
Draft & Print (Konami)	G1301	$85
Easy CAD 2 (Evolution)	G1214	$135
Instant Artist (Autodesk)	G1303	$59
KeyCAD Complete (Softkey)	G1307	$51
New Print Shop (Broderbund)	G1211	$43
PC Paintbrush IV (Z Soft)	G1212	$75
PC Paintbrush V+ (Z Soft)	G1216	$107
Presentation Task Force Clip Art (New Vision)	G1304	$145
Print Shop Deluxe (Broderbund)	G1217	$63
Professional Draw for Windows (Gold Disk)	G1305	$289
TrueType for DOS (MicroLogic)	G1306	$55
TrueType - More Fonts for DOS	G1307	$81
Will Vinton's Playmation for Windows (CinePlay)	G1213	$299
Windows Draw! (Micrografx)	G1308	$109

VIDEOTAPES

Animation and General Entertainment

AmigaWorld's Animation Video Volume I	M1221	$19
AmigaWorld's Animation Video Volume II	M1222	$19
Animation 101 (Myriad Visual Adventures)	M1310	$24
Benny the Bear (Haas Animation) Great Children's Tape!	M1312	$14
Best of the Fests 1989 (PictureStart)	M1313	$24
Best of the Fests 1991 (PictureStart)	M1314	$24
Beyond the Mind's Eye Video (Miramar) New!	M1287	$19
Chronos Video (Miramar)	M1288	$19
Computer Visions Video (Pacific Arts)	M1289	$19
Green Toons (PictureStart)	M1317	$19
History of the Amiga	M1318	$15
Mind's Eye (Miramar)	M1319	$17
The Basics of BASIC (ViaGrafix)	M1320	$37
DeluxePaint Animate Video (Saddleback)	M1321	$24
Getting Started With OS/2, 2.0 (ViaGrafix)	M1322	$36

How To Shoot Super Videos Series from Adita Video

Know Your Camcorder (90 minutes)	M1323	$34
How To Shoot Video Like A Pro (90 minutes0	M1324	$34
Continuity and Combining Shots (90 minutes)	M1325	$34
Lighting Techniques & Recording Sound	M1326	$34
Basic Editing With Consumer Gear (90 minutes)	M1327	$34
Intermediate Editing with Prosumer Gear (90 minutes)	M1328	$34
Advanced Editing With Professional Gear (90 minutes)	M1329	$34
Adita Bundle - Get all 7 videos and save 25%!!	M1330	$175
Introduction to Paradox (ViaGrafix)	M1331	$45
Learning CorelDraw 3.0 (ViaGrafix)	M1332	$45
PC Virus: Understanding and Prevention (ViaGrafix)	M1333	$27
Productivity with OS/2 2.0 (ViaGrafix)	M1334	$45
Programming in C (ViaGrafix)	M1335	$37
Programming in C++ (ViaGrafix)	M1336	$37
Programming in Pascal (ViaGrafix)	M1337	$37
Programming in Visual Basic (ViaGrafix)	M1338	$45
Teach Yourself Windows 3.1 Video (ViaGrafix)	M1290	$24
Using Aldus Pagemaker 3.0 Video (M-USA)	M1291	$24
Using Borland's Quattro Video (M-USA)	M1292	$24
Using CorelDraw 3.0, Advanced (ViaGrafix)	M1339	$45
Using dBase IV Video (M-USA)	M1293	$24
Using dBase IV, Advanced (ViaGrafix)	M1340	$45
Using Excel 4.0, Advanced (ViaGrafix)	M1341	$45
Using Lotus 1-2-3 2.3 or 3.0 Videos (M-USA)	M1293/4	ea.$24
Using Lotus 1-2-3, Advanced, 2.4 or 3.1 (ViaGrafix)	M1342/3	$45
Using Lotus 1-2-3 for Windows Video (M-USA)	M1295	$24
Using Microsoft Word 5.5 Video (M-USA)	M1296	$24
Using Microsoft Word for Windows Video (M-USA)	M1297	$24
Using Microsoft Word for Windows 2.0, Advanced (ViaGrafix)	M1344	$45
Using MS-DOS 4.01 or 5.0 Videos (M-USA)	M1298/9	ea.$24
Using Paradox, Advanced (ViaGrafix)	M1345	$45
Using QuattroPro 4.0, Advanced (ViaGrafix)	M1346	$45
Using Windows 3.1, Advanced (ViaGrafix)	M1347	$51
Using WordPerfect 5.0 or 5.1 Videos (M-USA)	M1300/1	ea.$24
Using WordPerfect for Windows Video (M-USA)	M1302	$24
Understanding PC Networks Video (M-USA)	M1303	$24
Using Xerox Ventura Publisher 2.0 Video (M-USA)	M1304	$24
VideoMaker Video Volume I	M1236	$19
Videography (Peter Hitchcock Productions)	M1350	$32
WordPerfect 5.1, Secrets & Timesavers (ViaGrafix)	M1342	$36

HOT NEW GAMES--*JUST IN!*

Airline Transport Pilot (SubLogic)	E1200	$41
Amazon (Access)	E1201	$41
Ancient Art of War in the Skies (MicroProse)	E1202	$37
Battle Chess 4000 (InterPlay)	E1203	$37
Battles of Destiny (QQP)	E1204	$41
Campaign (Empire)	E1205	$33
Castles II (InterPlay)	E1206	$39
Conquest of Japan (Impressions)	E1207	$37
Cyber Empires (SSI)	E1208	$36
Discovery in the Steps of Columbus (Impressions)	E1209	$39
Dune II (Westwood Studios)	E1210	$45
Front Page Sports Football (Dynamix)	E1211	$43
Goblins (Sierra)	E1212	$29
Gods (Konami)	E1213	$29
Heimdall (Virgin Games)	E1214	$25
Humans (Gametek)	E1215	$31
Jump Jet (MicroProse)	E1216	$43
King's Quest VI: Heir Today (Sierra)	E1217	$49
The Lost Admiral (QQP) Strategy Game of the Year!	E1218	$39
Lure of the Temptress (Konami)	E1219	$41
Monopoly Deluxe (Virgin Games)	E1220	$37
Omar Sharif on Bridge (InterPlay)	E1221	$41
Pacific War (SSI)	E1222	$44
Quest for Glory III: Wages of War (Sierra)	E1223	$45
Shadow President (D.C. True Ltd.)	E1224	$45
Tristan Pinball (Amtex) Best Seller!	E1225	$41
Ultima VII Part 2: The Serpent Isle (Origin)	E1226	$49
Utopia (Konami)	E1227	$35
X Wing (Lucas Arts)	E1228	$40

ENTERTAINMENT SOFTWARE
ADVENTURE/SIMULATION

Aces of the Pacific (Sierra On-Line)	E1001	$49
Bane of Cosmic Forge II:Crusaders	E1005	$47
Bard's Tale III (Electronic Arts)	E1092	$39
Battle Isle (EA)	E1006	$37
B-17 Flying Fortress (MicroProse)	E1007	$41
Chuck Yeager's Air Combat (EA) New!	E1013	$39
Crisis in the Kremlin	E1015	$45
Darklands (MicroProse)	E1016	$47

Circle 111 on Reader Service Card

Dune (Virgin Games)	E1020	$37
Elvira II Jaws of Cerberus (Accolade)	E1021	$41
Eye of the Beholder II (SSI)	E1022	$41
F-117A (MicroProse)	E1024	$49
Gunship 2000 (MicroProse)	E1032	$45
Indiana Jones IV (Lucasfilm)	E1099	$39
Knights of the Sky (MicroProse)	E1037	$29
Lightspeed (Microprose)	E1252	$39
Lost Treasures of Infocom (Activision)	E1038	$44
Mantis: Experimental Fighter (MicroProse)	E1095	$47
Martian Memorandum (Access)	E1040	$39
Might and Magic 3 (New World Computing)	E1044	$37
Might and Magic 4 (New World Computing)	E1045	$43
Patriot (360/Electronic Arts) New!	E1097	$47
Police Quest III (Sierra On-Line) Best Seller!	E1051	$43
Realms (Virgin Games)	E1052	$24
Second Front: Germany Turns East (SSI)	E1057	$39
Secret Weapons of Luftwaffe (Lucasfilms) Best Seller!	E1058	$45
Shuttle (Virgin Games)	E1060	$34
Siege (Electronic Arts)	E1061	$43
Silent Service II (MicroProse)	E1062	$37
SpellJammer Game World: Pirates of Realmspace (SSI) New!	E1098	$45
The Battle of Britain/Battlehawks (Lucasfilm)	E1069	$43
Ultima VII (Origin) Best Seller!	E1071	$49
Wizardry Trilogy (Sir-Tech Software)	E1074	$33
Wing Commander II (Origin)	E1075	$49

ENTERTAINMENT SOFTWARE ACTION/ARCADE/TRADITIONAL

A-Train (Maxis)	A1154	$43
American Gladiators (GameTek) Best Seller!	A1077	$27
Armour-Geddon (Psygnosis) New!	A1153	$37
Chessmaster 3000 (Software ToolWorks)	A1155	$39
Civilization (MicroProse) Best Seller!	A1083	$44
Corporation (Virgin Games)	A1084	$25
Global Effect (Millennium)	A1088	$39
Grand Slam Bridge II (Electronic Arts) New!	A1156	$39
Heaven and Earth (Buena Vista Software)	A1089	$34
Leisure Suit Larry 5 (Sierra On-Line)	A1090	$45
Lemmings (Psygnosis)	A1091	$29
Lost Files of Sherlock Holmes (Electronic Arts) New!	A1157	$41
Oh No! More Lemmings (Psygnosis)	A1093	$34
Populous (EA)	A1094	$22
Populous II (EA)	A1158	$39
Power Monger (EA)	A1095	$39
Railroad Tycoon (MicroProse)	A1097	$27
Risk (Virgin Games) Best Seller!	A1098	$29
SimAnt (Maxis) Best Seller!	A1100	$39
Sim City (Maxis) Best Seller!	A1101	$34
Sim Earth (Maxis) Best Seller!	A1102	$45
Solitaire's Journey (QQP)	A1103	$41
Speedball 2 (Konami)	A1106	$41
Star Trek 25th Anniv. (InterPlay) Best Seller!	A1108	$39
Super Tetris (Spectrum Holobyte)	A1109	$35
Wordtris (Spectrum Holobyte)	A1113	$29

ENTERTAINMENT SOFTWARE SPORTS

Earl Weaver Baseball II (EA)	A1118	$29
Greens: Ultimate Golf Simulation (MicroProse) New!	A1163	$41
HardBall III (Accolade)	A1164	$39
Hole-in-One Miniature Golf (DigiTek)	A1122	$22
Jack Nicklaus: Signature Edition (Accolade)	A1165	$44
John Madden Football II(EA)	A1126	$39
Jordan in Flight (EA)	A1127	$47
Links (Access)	A1128	$39
Links 386 Pro (Access)	A1166	$44
MicroLeague Football:Deluxe Edition	A1131	$44
Pro Football Analyst (Villa Crespo)	A1134	$39
Road to Final Four (Bethesda Softworks)	A1136	$41
Test Drive III (Accolade)	A1138	$41
World Circuit Grand Prix Racing (MicroProse) New!	A1167	$44
3-D Pool (MicroProse)	A1140	$12

HARDWARE

Beetle Mouse - Your Choice, each with 2 colors	M1270	$35
Cable for modems	M1239	$11
Cable for printers	Call	$11
CHIC Cordless Mouse	M1271	$49
CHIC Trackball	M1272	$35
Complete (fax) Modem (The Complete PC)	M1401	$75
Complete (fax) Modem Plus (The Complete PC)	M1402	$105
Cordless Trackball	M1273	$99
Eliminator Gamecard	M1274	$34
Flight Stick	M1275	$55
Game Card III Automatic	M1276	$24
Game Port 2 Plus	M1277	$25
Jin Mouse	M1278	$25
Mousepads - box of 10	M1279	$49
Mousestick	M1280	$99
PC Trackball	M1281	$87
Pet Mouse (with Lemmings Demo)	M1282	$31
Sound Blaster (Creative Labs)	M1241	$109
Sound Blaster Pro (Creative Labs)	M1242	$209
Sound Blaster Speakers	M1243	$24
SupraModem 2400 - external	M1244	$89
SupraModem 2400 Plus - external	M1245	$129
SupraFaxModem 2400/9600 Plus	M1246	$159
SupraFaxModem V32 bis	M1247	$369
SupraModem 2400 internal	M1248	$69
SupraModem 2400 Plus internal	M1249	$105
Thrustmaster Joystick	M1283	$89
Thunder Board (Media Vision)	M1250	$95
Thunder Board for Windows 3.1 (Media Vision)	M1284	$139
VGAStereo F/X with 1MB (ATI Technologies)	M1400	$379
Videoblaster (Creative Labs)	M1285	$369
Warrior V Joystick	M1286	$15

BOOKS

Basic Programming: Inside & Out (Abacus) w/disk	B1270	$27
Batch File Powertools (Abacus) w/disk	B1271	$27
Color Publishing on the PC (Random House) w/disk	B1300	$41
Computer Viruses & Data Protection	B1285	$15
DOS 5.0 for Beginners (Abacus)	B1272	$15
DOS 5.0 Essentials	B1287	$15
Dr. DOS 6.0 (Abacus) w/disk	B1273	$29
Easy 1-2-3 For Small Business (Compute!)	B1301	$18
Hints, Maps and Solutions to Adventure Games (Compute!)	B1302	$16
Hollywood (Presentations) Tour Guide (Random House) w/disk	B1303	$45
InfoWorld Windows 3.1 Secrets (IDG Books) w/3 disks	B1353	$37
InfoWorld PC Secrets (IDG Books) w/2 disks	B1354	$37
The OS/2 Book (Abacus) New!	B1274	$19
Laser Printer Powertools	B1288	$24
Learning Lotus 1-2-3 (Compute!)	B1304	$20
Mastering Video Toaster Technology 2 disks New!	B1305	$39
Multimedia Erector Set (Random House) w/disk	B1306	$41
Official Book of King's Quest, 2nd Ed. (Compute!)	B1307	$16
Official Book of Leisure Suit Larry, 2nd Ed. (Compute!)	B1308	$14
Official Book of Police Quest (Compute!)	B1309	$14
Official Guide to Civilization (Compute!)	B1310	$14
Official Book of Ultima, 2nd Ed. (Compute!)	B1311	$18
Parents, Kids and Computers (Random House)	B1312	$20
PC Assembly Language: Step by Step (Abacus) 2 disk	B1275	$27
PC Intern	B1289	$44
PC World Excel 4 for Windows (IDG Books)	B1351	$29
PC World DOS 5.0 Complete Handbook (IDG Books) w/disk Best Seller!	B1352	$32
Power DOS (Random House)	B1313	$24
Pro Links: Official Guide to Links and Microsoft Golf	B1314	$18
QBasic for Beginners (Abacus) New!	B1276	$16
Quick and Easy Learning Lotus 1-2-3 (Compute!)	B1313	$14
Quick and Easy dBASE IV (Compute!)	B1316	$14
The '486 Book w/disk	B1290	$29
The Sound Blaster Book (Abacus) w/disk – New!	B1277	$31
Tips and Tricks for Your PC Printer	B1291	$24
Turbo C++: Step by Step (Abacus) w/disk – New!	B1278	$27
Turbo Pascal for Windows (Abacus) w/disk	B1279	$31
Turbo Vision Toolbox (Abacus) w/disk	B1280	$31
Unix System for MS-DOS Users (Compute!)	B1317	$21
Using Novell Netware Lite (Random House)	B1318	$20
Wicked Windows (Abacus) New!	B1281	$1
Windows 3.1 Complete (Abacus) w/disk	B1282	$2
Windows 3.1 Intern (Abacus) w/disk – New!	B1283	$
WordPerfect for Beginners (Abacus)	B1284	$1

HOT MANUFACTURERS OF THE MONTH

Car and Driver - The ten best cars...including Mercedes C11 and Porsche 959. Ten be tracks. You've read the reviews, now test drive the real things! (Electronic Arts) A1150 $

Challenge of the 5 Realms: Spellbound in the World of Nhagardia. State-of-the-art cinema effects and an intriguing story set in a complex and breathtakingly beautiful world. The late by Adventure Gaming Hall-of-Famer, Marc Miller. (MicroProse) E1081 $

F-15 Strike Eagle III - They've done it again with a revolutionary new visual system creating the most reali dogfighting ever available in a flight simulation. Needs '386 w/640K and DOS 5.0 (MicroProse) E1080 $

Rex Nebular and the Cosmic Gender Bender - Combines raucous space adventure and comedy with most dazzling and enjoyable graphics ever to enhance a PC screen. (MicroProse) E1082 $

Dark Sun - A quantum leap in computer AD&D role-playing on the most successful new fantasy world e launched - the Dark Sun game world! (Req. 640K, mouse & hard drive) (Strategic Simulations) E1083 $

SPECIAL! DEVWARE PRODUCTIVITY DISK

Best Seller--Super Value! Make every computing operation smoother and make yourse more productive when you employ Cruiser and the 14 other utility programs on th special disk. Cruiser allows you to see inside any file on your disk and page through as if the disk were one large book. Cruiser has a text editor, pop-up calculator, an much more! U1192 $

SPECIAL OFFER!

Our New Year's Gift To You--For every order over $200, receiv the DevWare Utility Disk Absolutely FREE! It's jam-packed wit 15 useful programs--a $20 value!!

NORTON UTILITIES 6.01 Rated a Best Buy by PC World readers for 1992! July 1992 PC World also rates its optimizer as a Best Buy! Best noted for its "world-class file recovery features." (Symantec) V1197 **$105**

NORTON BACKUP 2.0 FOR DOS Rated a Best Buy by PC World for Backup Software (July 1992). This program has a superb interface and makes fast, full backups (Symantec) V1196 $89

Return policy: All programs in origional packaging. Refunds only on product price. Call for Return Authorization Number.

ORDER BY TELEPHONE OR MAIL IN THIS FORM

ORDER FORM

Mail this order form to:
DEVWARE DIRECT,
12520 Kirkham Court, Suite 1-DR3
Poway, CA 92064

All games require 512K and DOS 2.2+.
Please specify disk size (5.25" or 3.5").

Shipping Charges
US: 3% of total order ($5 minimum)
Canada: 4% of total order ($6 minimum)
Foreign: 6% of total order ($10 minimum)
C.O.D. add $5 (U.S. orders only)
All payments in U.S. funds only.
Prices subject to change without notice.
30-Day money back guarantee on all products.

Name ____________________
Address ____________________
City ____________ State ____________ Zip ____________
Country ____________________
Day Phone #(   ) ____________________

Qty.	Product ID/Name	Unit Price	Total

7.75% Sales Tax (CA Res. only) ________
Shipping ________
Total ________

Method of Payment
☐Check/Money Order
☐Visa/MasterCard
☐Discover

Credit Card# ____________________
Expiration ____/____ Signature: ____________________

Order Toll-Free! (800) 879-0759

M-F 7am - 7pm P.S.T.
Sat. 9am - 5pm P.S.T.

Customer Service: (619) 679-2825
Fax your order to: (619) 679-2887

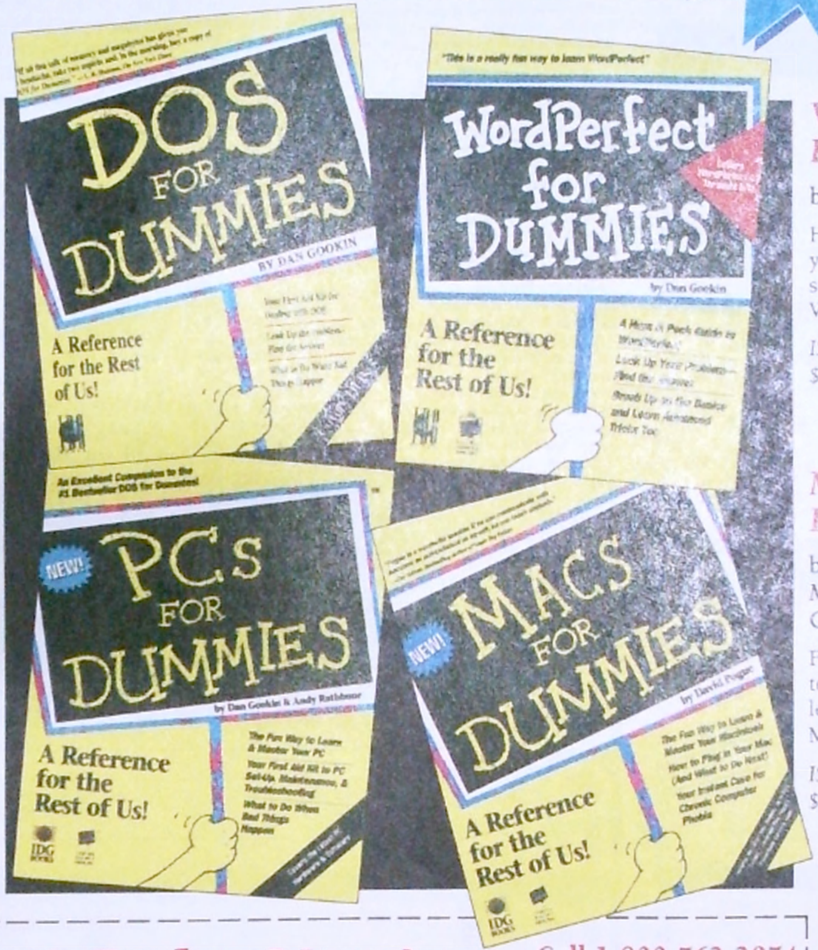
Circle 111 on Reader Service Card

Learn About Computers the Fun and Easy Way with IDG's *...For Dummies* Series

#1 Computer Books For Beginners!

Packed with useful information, yet written in everyday language, *...For Dummies* Computer Books™ explain the hows and whys of computers in a lighthearted and entertaining manner. Guaranteed to appeal to even the most reluctant computer user!

Only $16.95 Each!


DOS FOR DUMMIES
BY DAN GOOKIN
A Reference for the Rest of Us!
WordPerfect for DUMMIES
by Dan Gookin
A Reference for the Rest of Us!
An Excellent Companion to the #1 Bestseller DOS for Dummies!
NEW!
PCs FOR DUMMIES
by Dan Gookin & Andy Rathbone
A Reference for the Rest of Us!
NEW!
MACS FOR DUMMIES
by David Pogue
A Reference for the Rest of Us!

DOS For Dummies

by Dan Gookin

Blockbuster Bestseller! Explains DOS essentials with humor and style. Covers DOS 2 - 5.

ISBN: 1-878058-25-8
$16.95

PCs For Dummies

by Dan Gookin and Andy Rathbone

At last, a guide to PC configuration, upgrading, and repair that everyone can understand.

ISBN: 1-878058-51-7
$16.95

WordPerfect For Dummies

by Dan Gookin

Here are all the basics you need (and then some) for mastering WordPerfect quickly.

ISBN: 1-878058-52-5
$16.95

Macs For Dummies

by David Pogue, *Macworld* Magazine Columnist

Finally, a friendly guide to plugging in and learning how to use your Mac!

ISBN: 1-878058-53-3
$16.95

Get Smart! Order Now!

Call 1-800-762-2974
7 am to 5 pm PST or complete and mail this form.

Yes! Please send me:	Qty.	Price	Tax*	Ship	Total
________ *For Dummies*	______	$16.95	______	$4.00	______
________ *For Dummies*	______	$16.95	______	$4.00	______

UPS Shipping & Handling ($4 per book) *CA residents add applicable sales tax. IN residents add 5% tax. Canadian residents add 7% GST tax.

Name ____________________
Company Name ____________________
Address ____________________
City, State, Zip ____________________
Phone ____________________

Payment method (payment with credit card number or check must accompany your order.)
☐ Check enclosed ☐ Visa/Mastercard/American Express

Card # ____________________ Expiration Date ________
Signature ____________________

Inquire about rush delivery service (an extra charge will apply).
Corporate trainers quantity discounts available.

DRGDUM

IDG BOOKS
The World of Computer Knowledge

IDG Books Worldwide
155 Bovet Road, Suite 610
San Mateo, CA 94402
(415) 312-0608
(800) 762-2974

Circle 104 on Reader Service Card

SUBSCRIBE TO DOS RESOURCE GUIDE 800-238-6580

PC Essentials For Everyday Users
DOS RESOURCE GUIDE
Make It Work For YOU!
Speed Up Your Hard Drive
Money Saving Batch Techniques
42 More Great Tips Inside!
GUIDE TO WINDOWS SOFTWARE
BEST PROGRAMS FOR:
THE ESSENTIAL GUIDE TO DOS
2nd Edition
DOS 5.0
Batch File Tips and Tricks
MANAGING MEMORY STEP-BY-STEP
COMPLETE GUIDE To DOS Files
HOW TO TAME YOUR HARD DRIVE

SAVE UP TO 25% OFF

DOS RESOURCE GUIDE
THE COMPLETE DOS Q&A:
7 WAYS TO SAFEGUARD YOUR DATA
THE ONE-STOP DOS PRIMER
The 10 Worst DOS Commands
PROTECT YOUR DATA FROM VIRUS ATTACK
Peter Norton's COPY Secrets
13 Ways To Boost Your DOS 5.0
GET CREATIVE WITH DOS!
Cheap Investment Pays Big Dividends
15 DOS Tips

THE COVER PRICE

BUCK THE TREND! STAY WITH DOS

Get 6 issues of DOS RESOURCE GUIDE for $23.70.* That's just $3.95 an issue!

Save even more on 12 issues: $44.00.* That's 25% off the cover price!

Understanding DOS is the key to effective PC-computing. But if you've tried to learn DOS on your own, wrestled with its manual, purchased a guidebook, or taken a course, you're probably still frustrated and confused.

Relax, you're not alone!

One estimate puts the number of DOS-run computers in the marketplace at 70 million. But how many DOS experts are there?

Darn few, and here's why.

Since late 1980, Microsoft Corp.'s DOS, the essential software ingredient of the IBM PC-compatible, has changed many times. Along the way, the operating system that began as 22 cryptic commands has grown to about 100 equally cryptic commands, drivers, and programs.

DOS RESOURCE GUIDE explores DOS's hidden secrets, untangles the twisted technical prose of manuals, and tells you in clear language how to use DOS. In each issue of DOS RESOURCE GUIDE, you'll find articles about using DOS commands, about managing your computer's memory, or maintaining order on your hard disk.

Beginners will find step-by-step help at copying files, creating and removing directories, and practical advice on using DOS's wildcards to cut down on keystrokes.

More advanced users will learn about batch files, which can automate tedious activities, such as moving or backing up large groups of files. And you'll learn to use DOS 5's new QBasic programming language.

Each issue of DOS RESOURCE GUIDE is filled with tips on using DOS, undocumented shortcuts, and batch files you can type in and use.

So, tap all of your computing potential by learning DOS, personal computing's most important—if not its most popular— operating system. Order today! Use the coupon below, or call:
800-238-6580

☐ **YES!** Send me 6 Issues of DOS Resource Guide for $23.70.* Save me a buck an issue!

☐ **YES!** Save me EVEN MORE: I'll take 12 issues of DOS Resource Guide for $44.00.* That saves me 25% off the cover price!

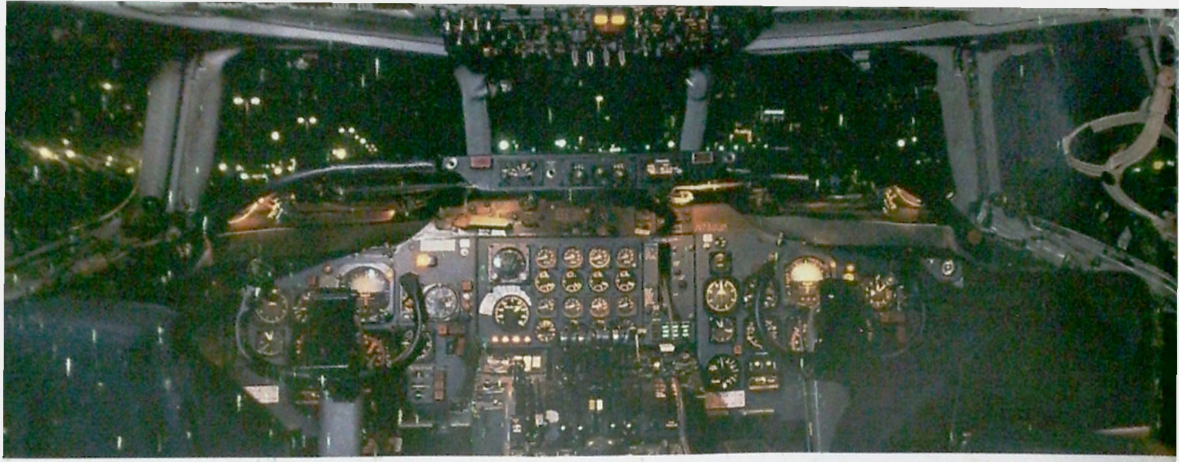
NAME ______________________________
ADDRESS ______________________________
CITY ____________________ STATE ________ ZIP ________

(Check, Money Order, or Credit Card—Payment must accompany your order.)
☐ AMEX ☐ VISA ☐ MASTERCARD ☐ DISCOVER
CREDIT CARD # ______________________________
EXPIRATION DATE: ______________________________
SIGNATURE: ______________________________

MAIL TO: DOS Resource Guide, 80 Elm Street, Peterborough, NH 03458

*In Canada and Mexico pay $29.70/(12 issues $55.00); all other foreign surface mail pay $41.70/(12 issues $78.00); foreign air mail pay $53.70/(12 issues $99.00). Checks and money orders must be payable in U.S. funds, drawn on a U.S. bank. Maryland residents add 5% sales tax. Please allow 6—8 weeks for delivery.

HVSAD



Get the best of Windows and OS/2 without learning all this.

Sure, you want to get your PC flying. With features like multiple open applications and background printing. Maybe even the ability to maintain your host connection while you work on something else.

But for that you need Windows or OS/2. Which means you also need to learn enough to get a pilot's license. And spend enough on hardware and software for a down payment on a jet.

PC MAGAZINE
EDITORS' CHOICE
October 30, 1990
Software Carousel, Version [illegible]

But with new Software Carousel, you can get the biggest benefits of Windows and OS/2 by doing almost nothing.

It's #1 in task switching software.

Software Carousel got to be a bestseller because it works. Simply and reliably. Day in and day out.

And because it gives you the single most practical and powerful function of OS/2 and Windows—the ability to quickly switch from one application to another.

Simply load your programs as usual in individual Software Carousel work areas (it handles up to 12). Then switch instantly from one to another with just a keystroke. Each program gets as much RAM as it needs. And all your programs look and act just like they always did.

Best of all, nearly any program that now runs on your PC will work with Software Carousel. Even RAM resident utilities, graphics programs and network software. And Software Carousel works on all types of PCs. So you can have the kind of multi-application capability you want, without buying anything new.

Now with Print'N'Run.

Thanks to Print'N'Run, new Software Carousel is also an advanced print handler. One that quickly takes over your printing jobs by accepting all the output bound for the printer, then sending it to the printer as fast as it can take it.

While your printer keeps running, you can keep working in the same application. Or switch to another Software Carousel work area running a different application.

No more wasted time waiting for your printer. No more lost productivity.

OLÉ. A network idea whose time has come.

OLÉ is the optional Open Link Extender* for Software Carousel. And it could be the best thing to happen to networks since OS/2 itself.

With OLÉ and Software Carousel, you can connect to any host or service over a network—whether it's a mainframe, a mini, etc.—then "switch away" to work on another application, and not lose your connection.

OLÉ even makes sure that incoming data is received, even though your PC may be occupied with another application.†

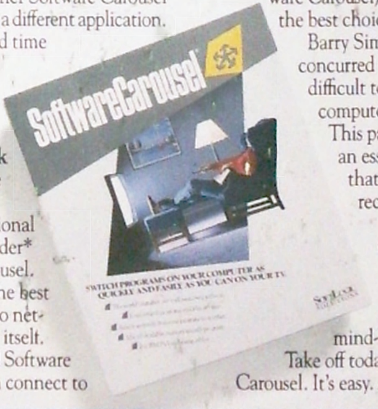
And OLÉ works with all kinds of connection software. Including IBM, Attachmate, and others. Without changes to your hardware or software.

Even the experts agree.

Garry Ray, writing for PC Week, said, "Of these alternative operating environments (OS/2, DesqView and Software Carousel), Carousel may be the best choice of the day."

Barry Simon of PC Magazine concurred with, "...I find it difficult to imagine using my computer without Carousel. This package has become an essential tool and one that I strongly recommend."

So if you really want the major benefits of Windows and OS/2, don't get grounded with high cost and mind-bending complexity. Take off today with Software Carousel. It's easy. And it's just $89.95.


SoftwareCarousel
SWITCH PROGRAMS ON YOUR COMPUTER AS QUICKLY AND EASILY AS YOU CAN ON YOUR TV
SoftLogic Solutions

Software Carousel $89.95

SoftLogic Solutions, Inc.
One Perimeter Road, Manchester, NH 03103

Available at software dealers everywhere.
Or, order direct by calling toll-free.
Call 800-272-9900
(603-627-9900 in NH)

SOFTLOGIC
SOLUTIONS

Software Carousel, OLÉ, Open Link Extender and Print'N'Run are trademarks or registered trademarks of SoftLogic Solutions. Other brand names are trademarks or registered trademarks of their respective owners. © 1990 SLS.

*Open Link Extender is sold separately. †Requires NetBIOS connection.

Circle 105 on Reader Service Card